Faculty of Computing Sciences and Engineering

Information Structuring and Reuse for Virtual Manufacturing

Thomas Karlsson

A dissertation submitted in partial fulfilment of the requirements for the degree of:

Doctor of Philosophy

October 2005

Sponsoring Establishment:

Abstract

This thesis presents a research investigation undertaken to enable reuse of data and models for virtual manufacturing with a special interest in discrete event simulation. The thesis argues that reusing data and models can save time and money and thus potentially offers advantages when reconfiguring existing manufacturing equipment. The investigation addresses key issues such as minimising modelling time and data collection. For the modern manufacturing/engineering industry, the major benefit is the possibility of rapidly identifying those resources along a production line that must be given new capabilities in order to manufacture a new product.

Manufacturers today show considerable interest for computer-based simulation tools as an aid in the process of developing production systems. Reports from industry have in some cases shown significant savings through the use of these tools. However, one of the disadvantages of the expanding use of simulation is the traditional one-to-one relationship between models and systems that seems to be prevailing.

Simultaneously, information technology has evolved over the last few decades at an extremely rapid pace. For instance, the internet technology has changed the way in which many businesses operate. Yet the true potential of information technology lies in its ability to connect and share processes and information irrespective of time and location.

This thesis explores the formal structuring of information and data required to conduct a simulation study with the possibility of reusing data and information. This suggests that data and information from previous studies can be reused, and likewise, future studies can utilise data and information from current studies.

A framework that enables the reuse of data and models is proposed. The framework combines virtual manufacturing tools and work methods with information management theories. The combination utilises international standards to provide a set of rules for sharing information in the framework.

The formulated framework has been tested in two industrial studies. One study reinforced the architectural design of the framework and identified data dependencies. The other study successfully tested the formulated framework on a full scale manufacturing unit.

This research successfully advances simulation data and model reusability in the area of virtual manufacturing with a focus on discrete event simulation. Furthermore, the research demonstrates the successful use of standards together with off-the-shelf simulation packages in a framework.

Acknowledgement

I would like to express my sincere gratitude to my supervisors Professor Philip R. Moore, Professor Leo J. De Vin and Dr Jan Oscarsson for their patience and encouraging guidance through my PhD studies. I would also like to thank the University of Skövde for financial support during my research.

A special thanks to my friends and family whom have supported me during the whole time, Thank You!

Finally I would like to dedicate this work to my wife Laura who always believes in me.

Thomas Karlsson

Skövde 2005

Table of Content

List of Figures

List of Acronyms and Abbreviations

3D	Three Dimensional
ADO	Active X Data Objects
AGV	Automated Guided Vehicle
AIM	Application Interpreted Model
ANSI	American National Standard Institute
AP	Application Protocol
API	Application Programming Interface
ARM	Application Reference Model
ASC	Accredited Standards Committee
BCL	Batch Control Language
C	ANSI C Programming Language
CAD	Computer Aided Design
CALS	Continuous Acquisition and Life-cycle Support
CAM	Computer Aided Manufacturing
CAPE	Computer Aided Process Engineering
CAPP	Computer Aided Process Planning
CAx	Computer Aided X
CIM	Computer Integrated Manufacturing
CIMOSA	Computer Integrated Manufacturing-Open Systems Architecture
COM	Component Object Model
CORBA	Common Object Request Broker (OMG)
DBMS	DataBase Management System

DCOM	Distributed COM
DES	Discrete Event Simulation
DM	Data Management
DOE	Design Of Experiments
EDI	Electronic Data Interchange
ER	Entity Relationship
ERP	Enterprise Resource Planning
EXPRESS	Information Modelling Language (STEP)
EXPRESS-G	Graphical Information Modelling Language (STEP)
FEA	Finite Element Analysis
FORTRAN	Procedural Programming Language
GERAM	Generalised Enterprise Reference Architecture and Methodology
GIM	GRAI Integrated Methodology
GRAI	Reference architecture developed at the University of Bordeaux
HS	Högskolan Skövde
HTML	HyperText Mark-up Language
IDL	Interface Definition Language
IS	Information System
ISO	International Standardisation Organisation
IVF	Industriforskning och Utveckling AB
KTH	Kungliga Tekniska Högskolan
LITH	Linköpings Tekniska Högskola
MANDATE	The Manufacturing Management Data Standard

MBS	Multi Body Simulation
MDM	Manufacturing Data Management
NIIIP	National Industrial Information Infrastructure Protocols
NIST	National Institute of Standards and Technology
OLE	Object Linking Embedded
OMG	Object Management Group
ORB	Object Request Broker
PC	Personal Computer
PDM	Product Data Management
PERA	Perdue Enterprise Reference Architecture
PLC	Programmable Logic Controller
PPR	Product Process Resource
RMSD	Information integrator for Reusable Manufacturing Simulation Data
RS	Robotic Simulation
SC	Scientific Committee
SCL	Simulation Control Language
SDAI	Standard Data Access Interface
SGML	Standard Generalised Mark-up Language
SQL	Sequence Query Language
STEP	Standard for the Exchange of Product Model Data
TDM	Technical Data Management
US	United States
W3C	World Wide Web Consortium

VCC	Volvo Cars Corporation
VCE	Volvo Construction Equipment
WG	Working Group
WIP	Work In Progress
VM	Virtual Manufacturing
VMS	Virtual Manufacturing System
VR	Virtual Reality
VRML	Virtual Reality Mark-up Language
WSC	Winter Simulation Conference
XML	eXtensible Mark-up Language

1 Introduction

The manufacturing industry is continuously facing new challenges. Post-war industry has seen the emergence of NC, 3D CAD, and a transformation from make-to-stock to make-to-order, just to name a few examples. The current manufacturing landscape is thus characterised by frequent product changes, a need for cost reduction as well as for increased flexibility, a continuous grouping and re-grouping of supply-demand networks in a global market place.

One of the challenges for manufacturing industry today is how to accommodate product changes (variants, new products, or a change in product mix) at short notice. Manufacturing systems are required to be adaptable and upgradeable, and one way to achieve this is through modularisation [Steiner et. al., 2002]. However, whilst modularisation facilitates changes in a manufacturing system, there is also a demand to reduce the time required to make these changes and to minimise any disruption of production. As an example, testing needs to be reduced to an absolute minimum and preferably eliminated altogether ("plug & produce" instead of "plug and debug"). This means that changes must be planned in detail and their consequences analysed thoroughly before they are implemented, all within a very limited time-frame. Having the right information available at the right time thus becomes a crucial factor.

1.1 Background

The term *virtual manufacturing model reuse* can imply various things from the reuse of small portions of code, through component reuse, to the reuse of complete models. When considering the models of manufacturing systems in particular, their reuse is directly linked to a strong economical interest since they often represent physical production lines. Thus the reuse of a manufacturing system model also demonstrates the reuse of the physical environment it represents.

At a more abstract level, component design, model design and modelling knowledge are prime candidates for reuse. The reuse of virtual manufacturing models[1] and its data is especially appealing, because it is based on the intuitive argument that it should reduce the time and cost of model development. However, a number of issues have arisen which indicate that these benefits may not be obtainable. These issues include the motivation to develop reusable models, their validity and credibility as well as the cost and time required for familiarisation.

1.1.1 Reuse, Extensibility and Evolveability

Parallels can be drawn between virtual manufacturing models and software components. Virtual manufacturing models largely consist of a code programmed by an engineer in an application specific environment in order to provide functionality to a model. Software components are developed under similar circumstances with similar requirements. Even the same development methodology can be used in both cases. Therefore, advancements achieved in the area of software development can to some extent, be used in virtual manufacturing development, for example: reuse, extensibility and evolveability.

Software reuse can vary. At one extreme in an idealised plug and-play world, components are used in an executable form and plugged together to produce an application. Java programs work this way, though it is normal to distribute their source code as well. At the other extreme are situations in which the source code is available and can be modified. That is, existing components may form the basis of customised components that are then linked to form an application. With such an approach, components can be extended with new functionality as required, and then reused.

[1] Virtual manufacturing models are a digitalised model of a part or a whole manufacturing system. Analyses that can be performed with virtual manufacturing are for example, robot simulation with off-line programming, ergonomical studies and logistic studies.

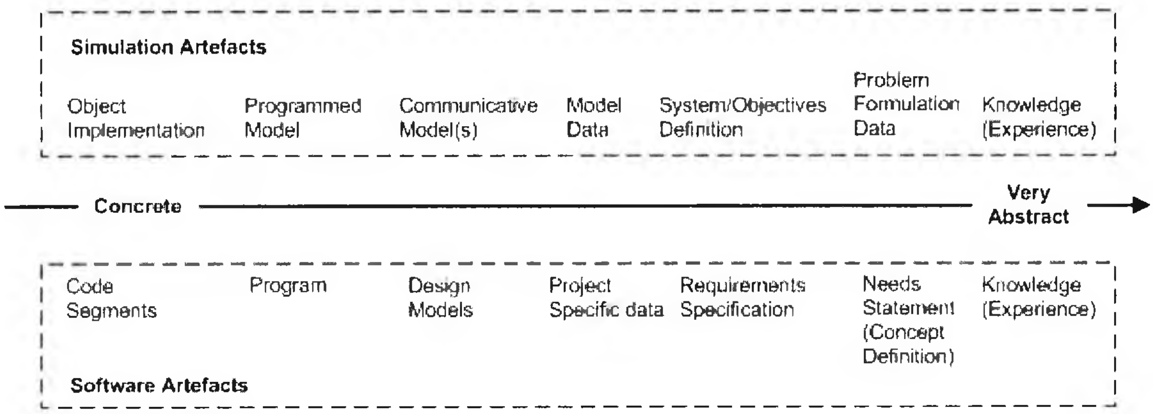

Figure 1 The abstraction continuum for simulation and software [Nance, 1994]

As well as the source code, other elements of a system can be reused, including those from the requirements specification, the design, its implementation and its testing. In fact, software reuse includes the isolation, selection, maintenance and utilisation of existing software artefacts in the development of new systems [Reese and Wyatt, 1987]. This is clearly a much broader application of reuse than just source codes or pre-compiled components. However, much previous research comes from computer science and has been focused on the link between essential software components and software architectures see Figure 1.

Although reuse and the consequent increase of productivity are important, the main initiative for component technology is that systems become extensible and evolvable [Szyperski, 1999]. An extensible system is one to which new parts that will be able to interact with the existing parts of the system, can be added. Systems are evolvable if existing parts of the system can be replaced by new components to improve their quality or features, whilst retaining the same underlying functionality. The obvious analogy is computer hardware that is both extensible and evolvable through the addition of new components, e.g. an Ethernet card, the improvement of other parts, e.g. video cards, whilst the other elements of the computer still work. For example, the parts will interact when a web page is accessed via a LAN through the Ethernet card, and fast moving graphics that use the video card will be seen.

Why is it necessary for virtual manufacturing systems in general to be extensible and evolvable? The answer is simple; the requirements of users change rapidly and virtual models must change to meet these challenges [Robinson et. al., 2004]. Virtual manufacturing models must be designed so that they can be extended and evolved to meet changing requirements. Furthermore, organisations invest much money and resources building models, and it may be unwise to completely discard them when extending the life of these investments makes commercial sense. Hence the enthusiasm for approaches which avoid the necessity of starting from scratch each time.

1.1.2 What is Simulation Model Reuse?

Consider simulation engineers who develop computerised models using a commercial-of-the-shelf (COTS) simulation package. Each package typically contains a set of predefined components that represent entry/exit points, queues, workstations, resources and entities. New models are built by combining these to form an appropriate representation of the conceptual model. Examples include the identification of a bottleneck in a production line, or how to schedule personnel at an intensive care unit [Oscarsson and Urenda Moris, 2002]. In some cases, models can be built from more complex components that constitutes models previously developed elsewhere, i.e. these models are reused. Experienced simulation engineers have access to previously built models making it possible for these models, or parts thereof to be used to analyse analogous problems and systems, and/or to be adapted for use in different contexts. Similar arguments can be made for simulation engineers working in a modelling team who have access to a shared model library or for those COTS modelling packages that have libraries of modelling components.

An example is the case of the factory owner who was unsure how to increase production. He employs the services of a simulation engineer to help develop a strategy to accomplish this, and together they develop a conceptual model[2]. To implement this as a computerised model, the simulation engineer, employing model reuse has several apparent opportunities to save time building the model. These are [Robinson et. al., 2004]:

- *Reuse of basic modelling components.* The simulation engineer reuses the basic modelling components (workstations, resources, etc.) that are included in the COTS modelling package.
- *Reuse of subsystem models.* The simulation engineer uses previously developed models of various *generic* factory parts or ones accessed through a model library (a conveyor subsystem is often a good example of this) that can be adapted and used with a new model representing the factory. Alternatively, the factory owner might already have previously developed models of factory parts which he makes available to the simulation engineer.
- *Reuse of a similar model.* The simulation engineer has previously developed a model that has similar features to the factory being studied. The model is adapted appropriately.

The first of these, *reuse of basic modelling components* is performed by the simulation engineer selecting and using the modelling component. Experienced simulation engineers know this is not the full story. For example, the developers of a workstation component have some assumptions about how such element operate. A simulation engineer using this workstation in a model will have to test it in order to understand how it actually works in the COTS modelling package since there is no standard cross-package behaviour. When the simulation engineer uses the workstation component and it fails to appropriately model the particular machine, he can take advantage of programming facilities or links to other programs, included in most COTS packages, to customise the component. This implies that models thus built come with baggage, i.e. programmed behaviour and/or supporting

[2] A conceptual model is a model that describes the general functional relationship among components of a system. [Oxford English Dictionary, 2002] A conceptual model belongs to the early phases in a life cycle, when the specifics are still unknown to a large extent.

components required for the model to be simulated. Even worse is baggage that is extremely dependent on the version of the package, the platform being used, and even the way in which the operating system has been configured. The conclusion of *reuse of basic modelling components* is that they are reused only after testing and modification [Robinson et. al., 2004]. The original component often evolves significantly beyond its original form. Time is saved only if the original component fits into the system without major reconstruction.

In *reuse of subsystem models*, the simulation engineer identifies part of the factory that can be quickly modelled by reusing a previously developed subsystem component from the engineer's own library or the library of the modelling package he or she is using. Either way, the subsystem model must be tested to determine if it correctly models the subsystem and then modified appropriately. If this complex component has baggage, then these must also be checked and understood. This implies that unless a subsystem component is quite simple, a simulation engineer will have to spend a great deal of time understanding how the component works. Additionally, what is the likelihood of the subsystem component conveniently modelling the equivalent factory subsystem? In conclusion, for most cases, the reuse of a subsystem model could be more costly than developing it from scratch. Time is saved only if the subsystem models have properly documented input and output points. [Paul and Taylor, 2002]

Similar arguments can be made about *reuse of a similar model* where the thorough testing of the reused model will take longer than testing a subsystem component. It is possible to see a similar model, with appropriate modifications, being reused as the system it represents evolves. However, it is unlikely that an existing model will be capable of being used to model a similar system. For example, production lines appear similar in that they tend to be a linear series of buffers and processing stations. Will two production lines really be that similar when studied in detail? Would it be beneficial for a simulation engineer to start afresh rather than spending time attempting to establish how a similar model works and what modifications are required? Time, in this case, can be saved if the model is properly documented which helps the simulation engineer to decide if the existing model can be used as a base for the new model or if the new model must be built from scratch. [Paul and Taylor, 2002]

In the light of the above described, data reuse could be defined as data that is created for one purpose and is later used for another. Furthermore, model reuse could be defined as the use of a model which is created for one purpose and then reused for another without rebuilding the majority of it.

In the world of COTS simulation packages it is difficult to see practically how a model can be trusted without detailed verification and validation that may be more costly than developing the model from the start. In summary, what use is reuse then? The answer to this may well therefore be; no use. However, it is pleasing to note, that this is not always the case [Robinson et. al., 2004]. Several businesses are beginning to realise the cost savings model reuse might actually achieve if supported properly. The emergence of the concept of a best practice bureau has made it possible to organise the practice of simulation modelling within the organisation. This results in a set of models developed using common practices and terminology that can be reused within the context of organisation guidelines. It also means that verification and validation could initially be delayed and simulation model reuse could itself reduce modelling time and increase cost savings. A better answer to the question, what use is reuse, might be time and cost saving but under careful planning and management.

According to [Reese and Wyatt, 1987] software reuse is the isolation, selection, maintenance and utilisation of existing software artefacts in the development of new systems. Thus reuse can be productively applied to all stages of development [Pidd, 2002]. Figure 2 illustrates a spectrum of different types of reuse, cast in terms recognisable to the simulation community. The figure indicates that reuse is much more frequent on the right-hand end of the spectrum. The complexity line, running from right to left indicates that code scavenging is relatively simple, whereas successful reuse of an entire simulation models can be very difficult.

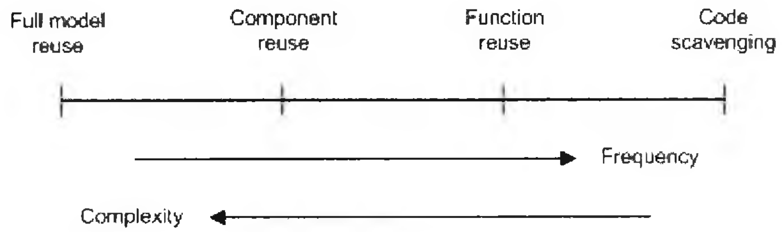

Figure 2 Spectrum of information reuse [Pidd, 2002]

What is frequently done is code scavenging, namely something that is known to work is modified into something new. If it is possible to a find code that can be reused, perhaps with some slight modifications, then this code will be used if the person who wrote the code in first place can be trusted. Function reuse is the next step along the spectrum. Pre-built functions are frequently used in the quest to solve a problem. Very little attention is focused on how the function works, which can be dangerous since not all functions operate as indicated, e.g. number generators.

There are many different definitions of a component, but it can be agreed that it is an encapsulated module with a defined interface [Pidd, 2002]. Components are commonly used in programming, where several functions are grouped inside a component. Further, it can also be a section of a simulation model which is grouped as a component. Full model reuse has been the holy grail in some parts of the simulation world, especially when models have been expensive and time-consuming to write and develop. A model may be reused many times for the same purpose, which is relatively straight forward. However, reusing a model for a different purpose or with a different configuration is clearly much more complex. By enabling such reuse many of the models built could be used to a larger extent which would lead to improved benefits on the effort put into the modelling.

1.1.3 Factors Affecting Product and Process Data Reuse

Trust is an important factor in data and model reuse. [Jacobsen et. al., 1992] argue that people do not always trust data collected and codes written by others, partly because they believe they have more control over their own data. They may also believe that they can implement, as well as use the code and data better. This may be justified, since large information systems often have a miss-match between their information models and the data they are handling [Kemmerer, 1999]. It is also easier to correct one's own faults than those of other people. Hence, proper testing, with correct documentation, is crucial if data and models are to be reused without modification [Oscarsson and Urenda Moris 2002].

It is a fact that poorly documented models are difficult to reuse. One reason why people want to see the source code is that it provides the detail of a model's operation. This desire is sensible, since for successful reuse a model must be well understood. The need for a full simulation code is obvious when a model is to be adjusted or extended.

Even if models exist, they must first be found which requires properly maintained libraries and directory services. [Sommerville, 2001] argues that some engineers do not regard the search for models as part of their job unless they are assured finding models or parts of models that suit their requirements. Though the purpose of models is to provide rapid development this does not imply that any simulation engineer, by using old models will find that the development process is facilitated, since he also requires expertise in reusing simulation models and data. This implies mastery of the architecture being used and significant knowledge of the models in the available libraries. If these requirements are not met, the promise of more efficient development will not be fulfilled through product and process data reuse. Proper training and specialised knowledge are required, as ever, to achieve full potential.

1.1.4 What Can Be Gained From Product and Process Data Reuse?

Somerville and Jacobsen suggest that frequently used models and data should, eventually, be of higher quality than traditional models because they will be thoroughly tested by continued reuse [Somerville, 2001] [Jacobsen et al 1992]. As a model is successively reused, bugs and/or weaknesses will be discovered by the simulation engineers and users, allowing its correction for future use plus understanding trends in behaviour. This implies that overall system quality increases, but also raises the question of the simulation engineer's responsibility for the usage of early model versions.

The development process for domain-based systems must be quite different from that of ordinary models, since there are two quite separate stages. Model development and model composition are usually carried out by different parties. Models are usually developed by domain experts, using a cyclic process in which models are developed and reviewed.

Systematic simulation model reuse does not just happen but must be planned, organised and encouraged [Jacobsen et. al., 1992]. As indicated above, model specification, implementation and testing are crucial for safe and straightforward reuse. This can mean that the initial development of a reusable model is more expensive than one that is not.

1.2 Research Aim and Objectives

The potential benefits and problems/limitations of the reuse of data and models has been briefly discussed. This constitutes both the motivation and point of departure for the research reported on in this thesis.

Furthermore, the work of Perera and his group [Robertson and Perera, 2002] clearly shows how information system integration with simulation systems can and should be achieved, see chapter 5. However, the studies and recommendations are based on only one type of information source, ERP systems. It is well known that there are many different types of information sources other than ERP systems which contain data useful for a simulation model. Therefore, the information systems integration with simulation systems must be broadened to cover other kinds of information sources with which a simulation system would have to interact.

As pointed out in the work from Kjellberg and his group [Kjellberg and Bohlin, 1996] standards and information modelling is a fruitful combination. The information in an information system automatically uses a clearly defined semantic by using ISO standards to represent them. When exchanging information between different systems, a common set of semantic rules are required. [Schenck and Wilson, 1994] Another vital aspect is functional structure, which, other than the importance of proper semantics, will work for human and computer interpretation [Kemmerer, 1999].

The primary research objective is to increase the reusability of the data that is used in simulation models for virtual manufacturing. An increased reuse of data for simulation models should preferably be supported by proper standards which will prove to fulfil the requirements of the simulation models. In order to increase simulation data reuse, this thesis attempts to answer the following question:

How can input and output data be organised using established standards and technologies in order to support simulation model reuse in virtual manufacturing?

This question can be further refined into the following objectives:

- Identify input and output data, standards suitable for the data that simulation models in virtual manufacturing require, as well as major systems that contain data used by the models.
- Formulate a framework that will support reuse of data for simulation models, based on the requirements and concepts that are identified.
- Implement and evaluate prototypes of the framework using modern techniques and tools, as well as systems used in industry today.

The Framework will support:

- The distributed management and storage of the data in a system that is based on commercial systems so that the user will feel familiar with both the user interface as well as the concept.
- Information management that is suitable for virtual manufacturing, in order to increase reusability of input and output data as well as models themselves.
- Information storage and structure of data within identified information domains with adequate semantics and provide context of the data that is put into the system.

1.3 Scope of the Work

The scope for this thesis is the reuse of data and information already known and captured in common types of information systems in industry. The information is normally scattered over several different information sources with often a very broad range of design intent and format. Therefore has some delimitation been set:

Standards: Only standards sanctioned by ISO will be used in this work. However, only use of the standard is considered and no development of them will be done.

Virtual Manufacturing domain: Concepts and methods are discussed and presented in relation to virtual manufacturing but exemplification is done using discrete event simulation as one tool in the domain.

Scientific approach: The problem area is studied mainly from an engineering and information management perspective.

Industrial Branch: Within the manufacturing industry the automotive sector is the most mature in Sweden. It is therefore natural to limit the scope to that particular industrial branch. There are several other sectors that perform discrete event simulation studies, e.g. the postal service and airports.

1.4 Thesis Organisation

Chapter one provides an overview of the thesis including the research background, research objectives, as well as organisation of the thesis.

Chapter two defines and describes virtual manufacturing systems and a methodology for building simulation models. The chapter also provides an overview of common integration techniques in virtual manufacturing.

Chapter three defines the differences between data, information and knowledge. It also describes the common information systems that provide most of the data for simulation. In this chapter standards, which are associated with information management for virtual manufacturing reviewed are also considered.

Chapter four explores the requirements of an information management framework suitable for reusing data for simulation models.

Chapter five describes the concepts of use and reuse of information for manufacturing simulations. It explains the building blocks of information and identifies information domains.

Chapter six describes the development and the developed framework in which reuse of simulation models are made possible. It explains how modern tool and techniques are used together with adequate international standards to achieve an environment that provides integrations between information sources and simulation tools.

Chapter seven describes two experiments that were conducted, one during the development of the framework and the other to test it.

Chapter eight concludes the thesis, the contribution to knowledge and recommendations for future work.

2 Concepts of Virtual Manufacturing

Virtual manufacturing is the name given to an area of research that aims to integrate diverse manufacturing related technologies. The scope can range from the integration of design sub-functions such as drafting, finite element analysis (FEA) and prototyping to the integration of all the functions within a manufacturing enterprise, such as planning operation and control [Shukla et. al., 1996]. Virtual Manufacturing (VM) is also identified as one of the enabling technologies of agile manufacturing and its related activities [Gunasekaran, 1999] Engineering activities related to manufacturing can be categorised into three main areas, namely i) products, ii) processes, iii) system [Oscarsson, 2000]. A wide range of simulation tools which support any of the design and operational management of these three areas are used in manufacturing.

2.1 Defining Virtual Manufacturing

Literature contains a wide variety of virtual manufacturing definitions and research directions. Some efforts focus on advanced visualisation of a new factory design using Virtual Reality (VR). Others aim at the integration of multiple organisations into one virtual enterprise for exploiting a market opportunity. Furthermore, other researchers suggest that VM is only concerned with the manufacturing activities that can be modelled and classified as production. One definition which is provided by [Iwata et. al., 1995] states that a Virtual Manufacturing system is an integrated computer based model which represents the physical and logical schema as well as the behaviour of real manufacturing systems.

Another similar description states that VM is a simulated model of the existing or non existing manufacturing setup. This definition also states that the model should contain all the information relating to the process, such as process control and management as well as product specific data. Furthermore, it should be possible that part of the manufacturing plant is real while other parts are virtual. The use of computer models and simulations of manufacturing processes to aid in the design and production of manufactured products is the basic purpose of Virtual manufacturing. [Onosato and Iwata 1993], [Kimura, 1993], [Nahavandi and Preece, 1994] and [Lin et. al., 1995]. Furthermore, virtual manufacturing is said to be an integrated, synthetic manufacturing environment which is constructed to enhance all levels of decisions and control in a manufacturing enterprise.

However, not everyone is satisfied with the definitions provided by Lin, Onosato and Kimura. Therefore, in 1996, a group of researchers and companies led by Lawrence Associates identified three different types of Virtual Manufacturing paradigms, depending on the context in which each is used during product development. These paradigms should not be implemented as overriding the definition provided by the other researcher, but should be regarded as a complement to the definition which captures the complexity of the VM concept. The three types of VM that can be distinguished are [Shukla et. el., 1996]:

- **Design-centred VM:** The use of simulation to optimise the design of a product and processes, and to evaluate many manufacturing scenarios at various levels of fidelity and scope in order to inform design and manufacturing decisions.
- **Production-centred VM:** The use of simulation during the manufacturing planning in order to evaluate and optimise manufacturing processes while taking into account resource availability.
- **Control-centred VM:** The addition of simulation to machine control models and actual processes, allowing for seamless simulation for optimization during the actual manufacturing.

The definitions provided by Lin, Onosato and Kimura together with the paradigms presented by the researchers led by Lawrence Associations, well capture the fact that VM uses modelling and simulation technologies as a tool to support design and manufacturing phases. However as stated by [Shukla et. el., 1996], the aim of VM is to "provide an integrated environment where concurrent manufacturing applications such as CAD, CAM and CAPP, can be carried out without physical constraints".

Virtual Manufacturing is concerned with creating virtual environments for the purpose of enabling a better concurrent engineering process. Virtual environments allow engineers from one activity e.g. designer, planner, to directly evaluate their impact of the decisions on other activities. Several types of VM applications are in use today in various phases of a company. Each of these provides a distinct part for the analysis of the system, which in its turn has an impact on another. In order to study each application, engineers need to develop models of the system. This is time and resource consuming and when problems cannot be solved by isolated models, two or more models from different systems have to be interconnected.

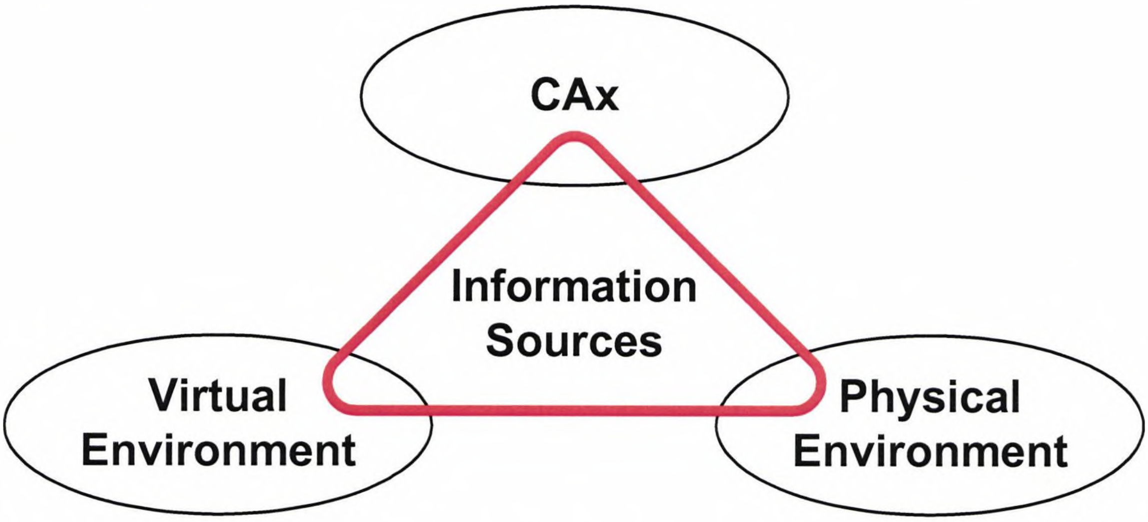

Figure 3 Virtual Manufacturing Concept [Bernard, 2000]

In the Virtual Manufacturing concept that is shown in *Figure 3*, a virtual environment allows several simulation applications to be processed concurrently by sharing data from common models and by integrating the applications. Moreover this virtual environment is integrated with information sources that provide information from CAx systems and from the physical system. With an available core information system, all applications are able to use the same accurate and updated information at the same time.

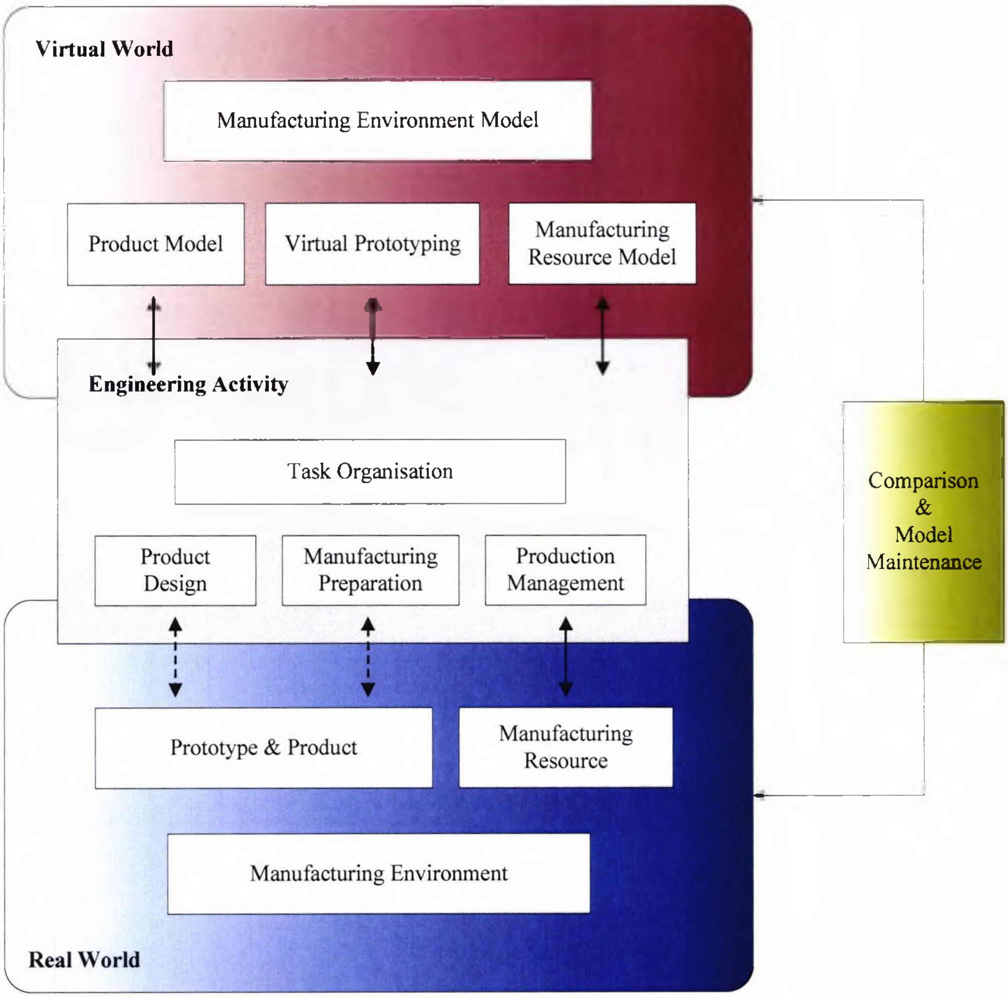

Figure 4 Concept of Virtual Manufacturing [Kimura, 1993]

Naturally, it is not straight forward when utilising VM in this broad perspective and traditionally, computer power and time have been the two main obstacles. To overcome those deficiencies and to utilise computer power more effectively Kimura proposed the following approach, see Figure 4 [Kimura, 1993]:

- The systematic organisation of manufacturing knowledge as much as possible based on relevant theory and accumulated knowledge.
- The comprehensive modelling of engineering objects and activities based on the above analysis.

- The evaluation of design and manufacturing activities based on precise computer simulations prior to real manufacturing.
- The elimination of inappropriate results by the previous evaluation.
- The maintenance of models in daily operation to achieve high-quality simulation.

This illustrates how VM should be used in a way that effectively utilises the resources at hand.

Various different definitions prevail with slightly shifted focus, e.g. the definitions of digital plant and virtual factory within the area of VM. However, in this thesis the definition of VM by Onosato, Iwata, Kimura, Nahavandi, Preece, and Lin is used.

To understand how close but still different some of the definitions are, Virtual Factory can be taken as an example. A virtual factory is an integrated model of major subsystems in a factory that considers the factory as a whole and provides an advanced decision support capability. The virtual factory normally seeks to go beyond the typical modelling of one sub-system at a time, such as the manufacturing model, the business process model and/or the communication network model developed individually and in isolation [Jain et. al. 2001]. Therefore, the virtual factory is within the scope for VM and at the same time a bridge to a slightly broader definition such as Digital Plant.

Definitions of Virtual Factory

Several definitions of the virtual factory exist in the application and research domain. These definitions have been broadly classified into four categories [Jain et. al. 2001]

1. *Virtual factory as a representation of all major aspects of the factory through one or several integrated simulation models.* This definition defines the virtual factory as a metaphor for the integration of a variety of software, modelling tools and methodologies which support solutions to a range of problems in the manufacturing domain.
2. *Virtual factory as a virtual organisation, i.e. partnering of multiple organisations for manufacturing a product.* [Upton and McAfee, 1996] define virtual factories as

"collaborative, internetworked environments in which several partners electronically share information an IS tools around a product (CAD/CAM, simulation-based design) process, or project". A model is presented that utilises information sharing tools developed for the Internet and rapidly increasing communications bandwidth to meet the demands of a virtual factory. Some of the solution providers offer services to develop virtual factories based on this definition.

3. *Virtual factory implementations emphasizing the virtual reality representation of a factory.* In addition to the visualisation and simulation, the capability allows quick access to drawings, procedures, databases, manufacturing applications and legacy information, captures process and equipment statistics and provides visual display of an integrated manufacturing knowledge base.
4. *Virtual factory as an emulation facility for production activity in a factory.* Projects based on this definition focus on modelling the production activity in a factory using simulation and emulation tools. The virtual factory emulator provides an interactive decision support system with user facilities to test operational, tactical and strategic decisions.

The view of a virtual factory as an integrated model of various sub-systems in a factory is similar to the one taken in some of the enterprise modelling architectures such as CIMOSA and GERAM [Vernadat, 1996].

2.2 Computer Simulation in Virtual Manufacturing

The modelling of a VM system, such as manufacturing systems, can be achieved by using a number of different tools and techniques. One family of these tools is computer based simulation. The Oxford English Dictionary describes simulation as [Oxford English Dictionary, 2002]:

> *The technique of imitating the behaviour of some situation or system (economic, mechanical, etc.) by means of an analogous model, situation, or apparatus, either to gain information more conveniently or to train personnel.*

Or put in another way, simulation is the technique of building a model of a real or proposed system so that the behaviour of the system under specific conditions may be studied [Ball 1996].

Since simulation is a general term misunderstandings can occur. Spreadsheet packages provide the possibility of generating *what-if* scenarios quite easily [Carrie 1988]. Ordinary programming languages and tools, such as MatLab with SimuLink, also allow the possibility for almost anyone to perform simulation. On the other hand there are numerous dedicated simulation tools, for e.g. FEM, RS, ergonomics, etc. that require proper training in the tools and an understanding of the area which is to be simulated. When using a simulation tool a system is captured in a model. The definition of a model provided by the [Oxford English Dictionary, 2002] is:

> *A simplified or idealised description of a system, situation, or process, often in mathematical terms, devised to facilitate calculations and predictions.*

Mathematical models are dominants and represent a system in terms of logical and quantitative relationships. That is manipulated and changed to illustrate how the model reacts, and thus how the system would respond provided the model is valid [Law and Kelton 1991].

Analytical solutions are exact solutions to the mathematical model. However, the majority of real world applications prove too complex and require vast computing resources. In those cases, the model must be studied by means of a simulation, i.e. numerically exercising the model for the inputs in question to see how they affect the output measures.

Models can be classified along three dimensions as suggested by [Banks, et. al., 1996], [Gogg and Mott, 1993] and [Law and Kelton 1991]:

- static or dynamic
- discrete or continuous
- deterministic or stochastic

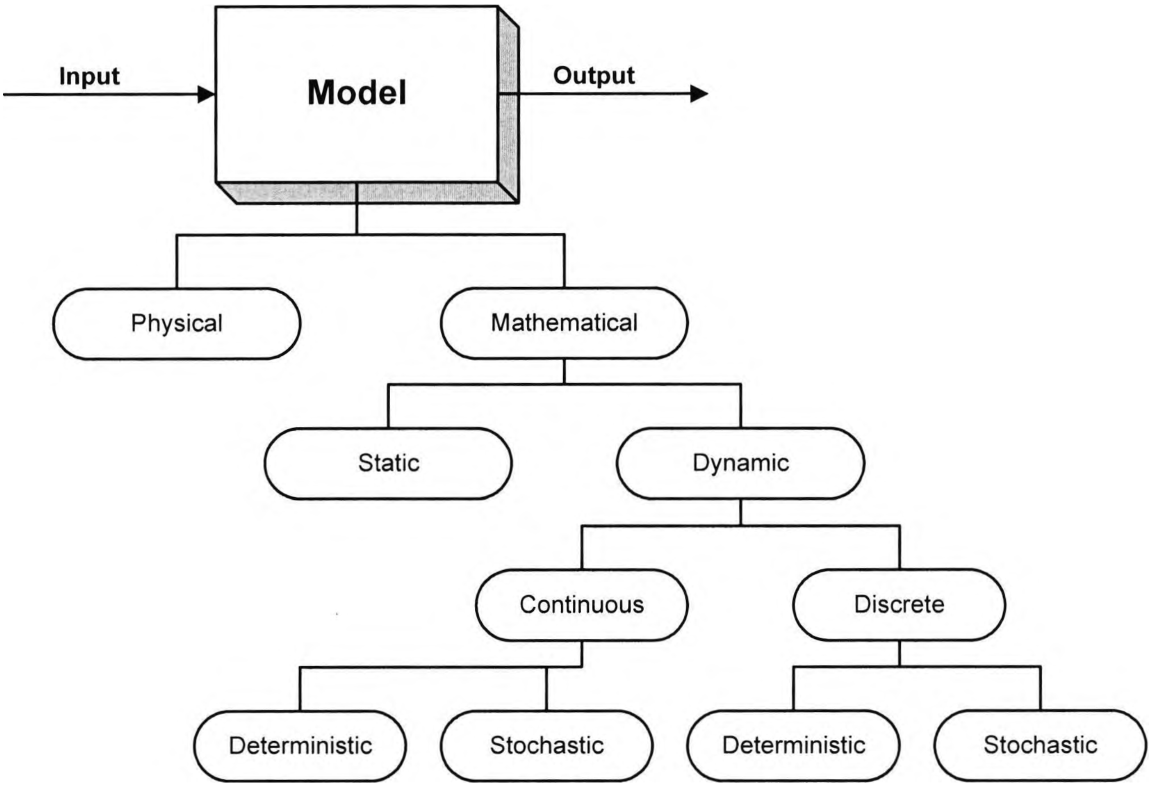

Figure 5 A model and its different analyses possibilities

A static simulation model is a representation of a system at a particular point of time. A dynamic simulation model represents a system over time, i.e. the system state, entity attributes and the number of active entities, the contents of sets, as well as the activities and delays currently in progress are all functions of time and are constantly changing over time.

If a model contains no probabilistic components it is considered to be deterministic, i.e. the result is always the same given the same input. However, in a stochastic simulation model the behaviour is determined by stochastic variables. In the real world most things are in fact stochastic. Typical stochastic variables in a manufacturing context are for example cycle times, mean time between failure (MTBF) and mean time to repair (MTTR).

A discrete simulation model is one in which the state variables change at a discrete set of times. The accuracy of discrete systems can be controlled by regulating the discrete sampling intervals. A continuous system is one in which the state variables change continuously over time.

[Banks et al., 1996] define a system as a group of objects that are joined together in some regular interaction or interdependence for the accomplishment of some purpose.

A system is often affected by the system's environment. It is therefore necessary to decide on the boundary between the system and its environment [Banks et al. 1996]. The boundary is highly dependent on the purpose of the simulation study.

The decomposition of a system into its entities simplifies the analysis. Although the studied system is large and complex, the entities are seldom complex. A logical model of a complex system may be built by incorporating a number of simple relationships, which provide the possibility of predicting the behaviour of the entire system. The more complex a system becomes the more simulation is preferred over theoretical equations [Carrie 1988].

To be precise we gain information only about certain aspects when simulating. Those aspects that have not been modelled can not be studied [Ball, 1996] [Carrie, 1988]. This is one of the fundamentals of a simulator; each is designed to serve some specific purposes. The definition of simulator from the [Oxford English Dictionary, 2002] is:

> *An apparatus for reproducing the behaviour of some situation or system; esp. one that is fitted with the controls of an aircraft, motor vehicle, etc., and gives the illusion to an operator of behaving like the real thing.*

Although simulation is classified into three domains by Law and Kelton the community considers simulation to be divided into a number of application specific technologies, which are [Law and Kelton, 1996]:

- **Discrete event simulation** is used in manufacturing mainly for the study of large scale production sections such as lines or whole plants.

- **Continuous Path Simulation** is used mainly for studying individual production cells or a limited area of production. Ergonomic studies are popular.
- **Process Simulation** is used mainly to study product related issues which helps the designer to validate a specific design, e.g. FEM and MBS

These domains derive from the specific types of tools used to help study and analyse a system. Within this thesis is only the discrete event simulation is considered and when referring to simulation, discrete event simulation is intended.

2.2.1 Discrete Event Simulation

Discrete event simulation is one method of building up models to observe the time based or dynamic behaviour of a system. There are formal methods for building simulation models and ensuring that they are credible. During the experimental phase the models are executed (run over time) in order to generate results. The results can then be used to provide insight into a system and a basis on which to make decisions. [Kreutzer, 1986]

2.2.2 Example Application Areas

There are a number of potential areas for the application of discrete event simulation. These application areas are introduced for the purpose of placing the use of this type of simulation into context. The range of application areas is extremely large and there are numerous examples of the use of simulation in service industries, manufacturing (batch and process) and office environments [Banks, 2000].

Simulation is commonly used in the area of developing new systems, particularly those that involve high capital investment. For example, simulation can be used to test the performance of personal computer (PC) assembly lines to ascertain the throughput possible, the level of utilisation of operators and any potential problems. Further, investigations can assess the best position for work-in-progress (WIP) stores and the levels to be used.

2.2.3 Key Principles of Discrete Event

The process of building simulation models will invariably involve some form of software. This software could either be a high level programming language or a data driven software system in which the model is specified utilising user-defined and default data items (e.g. machine scrap rate of 5% with default efficiency of 100%). Hence the model is either the software itself or it is held within a host software system. [Kreutzer, 1986] With the development of simulation systems it is generally the simulation engineer who will build the model. [Jägstam, 2004]

2.3 Using Simulation with a Methodology

There are different ways of studying a manufacturing system using simulation, as shown in Figure 6.

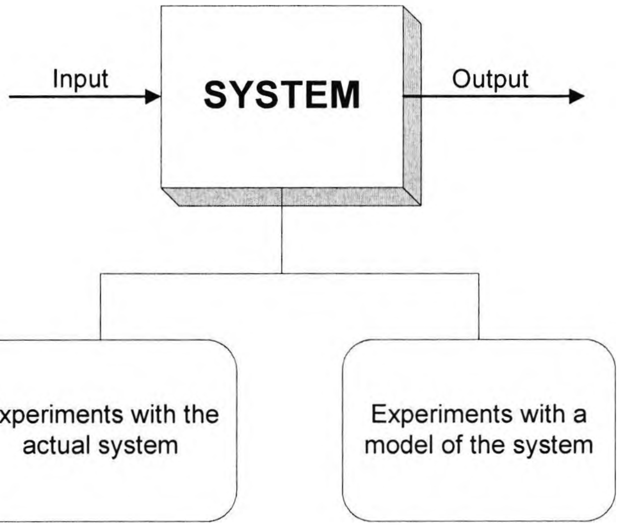

Figure 6 Alternatives in studying a system

In some rare cases experiments with the real system could be feasible. Furthermore, in cases where the system does not exist or experiments with the real system are associated with high costs, in lost production or investments, or if there is a time limitation, the study can be conducted on model of the system [Grewal et. al., 1998]. Three typical simulation study types can be identified:

- **The explorative study** to improve an existing system. The simulation model is used to rapidly make a number of changes to see if the system can be improved by e.g. changing the scheduling rules, operating rules, or the system itself
- **Study the existing system** with some suggested changes and comparing the results to see if the changes are profitable. This is similar to the previous study but here the model is used to validate proposed changes, not to find them.

- **To design and validate a system not yet in existence**. In this case the simulation is used in the design process to validate the performance or function of the system, and, at the same time, detect possible enhancements of the proposed system.

All model builders want their simulation project to be successful and useful for the user. To achieve the maximum output and to minimise mistakes, it is good practice to follow some guide lines or methodology in how to handle the simulation project and tackle the problems that arise while making the model. Through the years many systematic methods for conducting a simulation study have seen the light. Most of the methodologies use steps of activities which describe how to conduct a simulation study. There are many other methods for improvement that focus on the process instead. Vital activities involved in the development of a computer model are *model design, model execution* and *model analysis.* A simulation project often involves a number of people and to ensure its success, skills in both project management and simulation are necessary [Lee and Noh, 1997], [Law and Kelton, 1991], [Banks, 2000] and [Jägstam, 2004].

One methodology that is used and accepted in the community is the one developed by Banks [Banks et. al. 1996], which describes 12 separate steps the different activities in a simulation study in, see Figure 7.

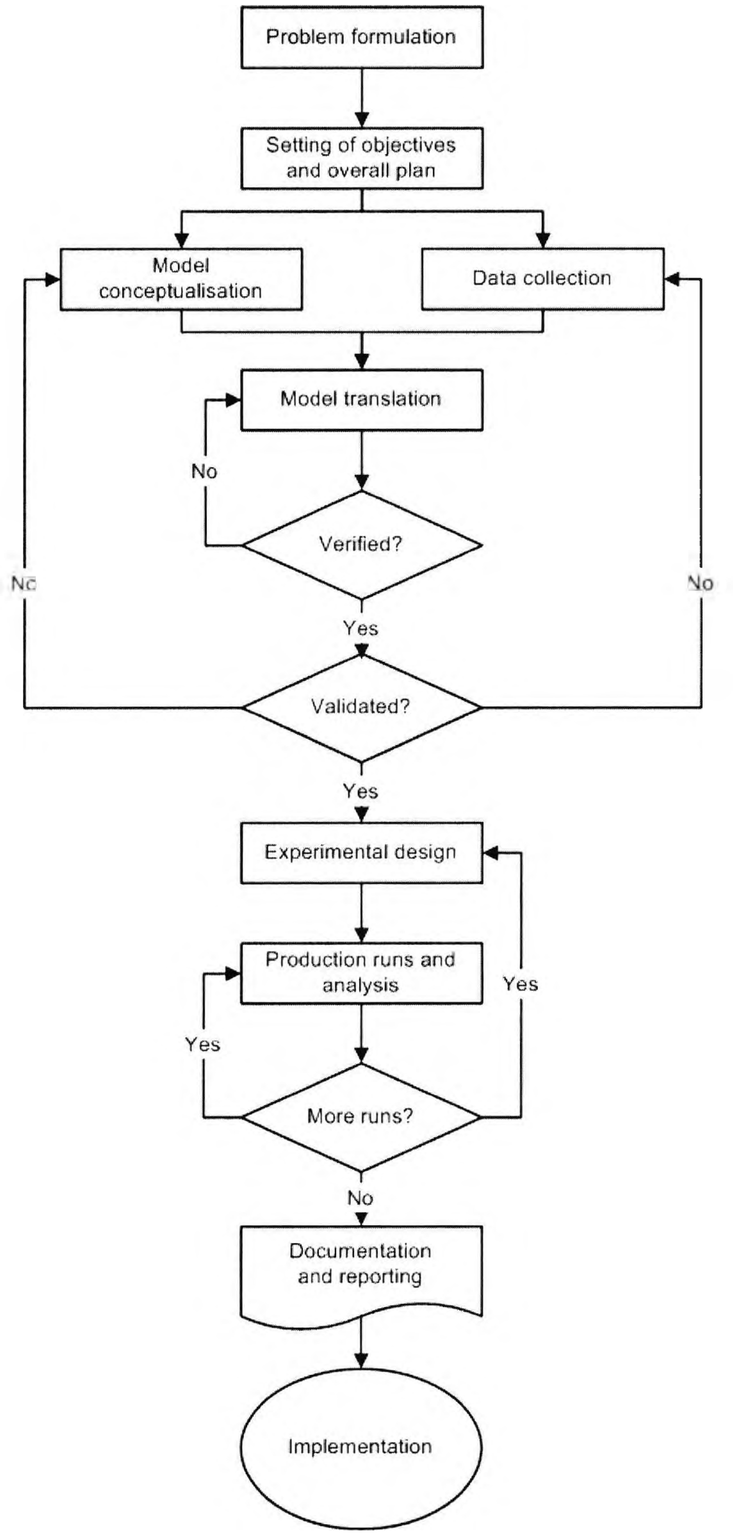

Figure 7 Steps in a simulation study [Banks, 2000]

2.3.1 Problem Formulation and Setting the Objectives

Shannon states that properly initiating a simulation study may be the decisive factor between success and failure. Every simulation study begins with a statement of the problem. If those with the problem provide the statement, extreme care must be taken by the simulation analyst to ensure that the problem is clearly understood. If the simulation analyst prepares a problem statement, it is important that the clients understand and agree with the formulation. It is suggested that a set of assumptions be prepared by the simulation analyst and agreed to by the client. Even with all of these precautions, it is possible that the problem will need to be reformulated as the simulation study progresses. This problem justifies questions such as [Shannon, 1998].

- What is the goal of the study, i.e. what are the questions to be answered or decisions to be made?
- What information do we need to make a decision?
- What are the precise criteria we will use to make the decisions?
- Who will make the decisions?
- Who will be affected by the decisions?

Despite rigorous preparations the formulation of the problem may need revising as the simulation study progresses.

Setting the objectives and planning the project should be accomplished, whether by an external or internal consultant, regardless of the location of the analyst and client. The objectives indicate the questions that are to be answered by the simulation study. The project plan should include a statement of the various scenarios that will be investigated. The plans of the study should indicate the time that will be required, personnel that will be used as well as hardware and software requirements. [Jägstam, 2004] has provided a handbook on project management for conducting simulation studies. It has shown that proper management of this kind of project is of great importance and that the distinctions between failure and success are mainly organisational questions.

2.3.2 Model Conceptualisation and Data Collection

The real world system under investigation is abstracted by a conceptual model. A series of mathematical and logical relationships concerns the components and the structure of the system. It is recommended that modelling begins simply and grows until a model of appropriate complexity has been developed. Constructing an exceptionally complex model will add to the cost of the study and the time for its completion without necessarily increasing the quality of the output. Maintaining client involvement will enhance the quality of the resulting model and increase the client's confidence in its use.

Following the concept's acceptance, data requirements should be established. In optimal circumstances, the required data has already been collected and can be submitted to the simulation analyst in electronic format. Often, companies claim that the required data is available. However, when the data is delivered to the simulation analyst, it is found to by quite different than expected [Sadowski and Grabau, 2000]. Therefore much effort has lately focused on data integration of simulation systems. The approach taken varies from different types of industries and continents.

2.3.3 Model Translation

The conceptual model that is constructed in the model conceptualisation phase is coded into a computer recognisable form. This is done either in simulation language software or in a special-purpose simulation package, simulator. Model translation is the step that seems to be the most time consuming. However, once the simulation language has been mastered the work is fairly straight forward. Further, without proper planning and analysis, the effort of this phase is useless. It is irrelevant how beautiful and detailed the model is, if it fails to address the original problem. It is important to build credible models, which in short means that the receiver of the model must trust it will fulfil its purpose [Sargent, 1998].

2.3.4 Verify

Verification concerns the operational model. Is the model performing properly? All the verification activities are closely related to the model and are aimed at analysing it to ensure so it acts according to the descriptions of the conceptual model. With complex models this can be very difficult and the modeller may need to verify the parts of the model separately. Even with a small model taken from textbooks, there is the possibility of verification problems. Consequently, verification should be a continuous process with the help of visualisation and code debugging [Balci, 1995]. An empirical method of checking if the model is behaving as intended is to put constant values for all the random processes in the model, and to calculate the output that the model is supposed to give. However, this is not sufficient or enough to verify a model. Verification is not complete until the model behaves as required according to the previously made conceptual model.

2.3.5 Validate

The validation process is mainly concerned with establishing the end user's confidence in the model [Balci, 1995]. Two typical questions need to be addressed during this stage:

1. Does the model adequately represent the relationships of the real system?

Validation compares the model built with the real system in order to justify the model's accurate representation of the real system. Due to the level of detail added to the model an exact representation will not be possible. In these cases validation is concerned with justifying how well the model represents the system. This is also the case when the real system is a non-existing one.

2. Is the model's generated output typical of the real system?

One way to validate a model of an existing system is to compare the model output (using historic input data) with the real output data from the same time span [Law and Kelton, 1991]. This method is called the correlated inspection approach, see Figure 8. If the model is a non-existing one, a person with experience of similar systems can justify the model. This method also allows the receiver of the model to look at both system data output and model data output and try to determine which one is real one.

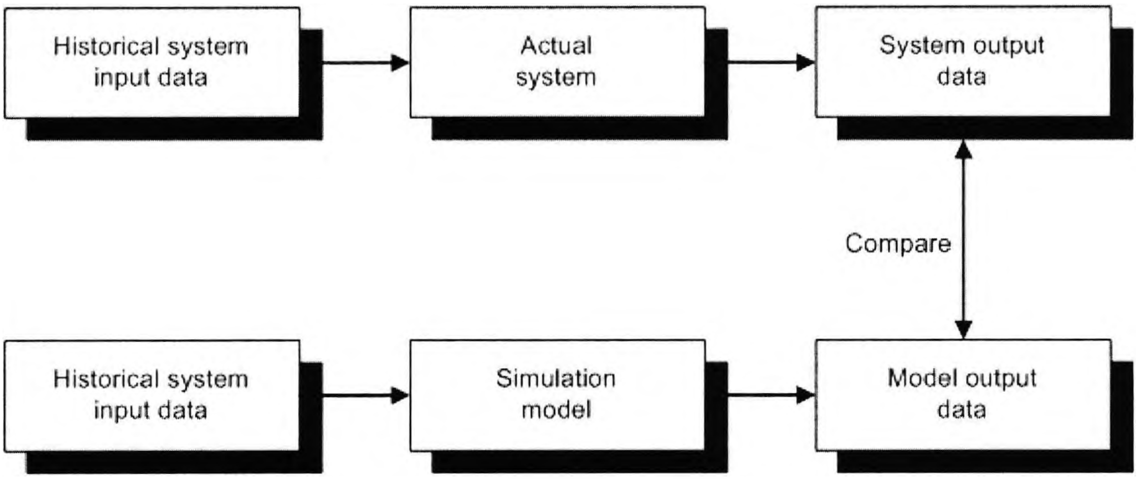

Figure 8 The correlated inspection approach [Law and Kelton, 1991]

Validation requires that models are tested for a reasonable amount of time to establish if they have continuity, consistency and degeneracy. A good way to start the validation process is to question the structural assumptions of the simulation model process.

2.3.6 Experimental Design

Since simulation in most cases contains stochastic data it is important to handle the runs in a way that statistically secures the accuracy of the output. This is achieved by analysing how long a model should run, how many replications are needed and how to handle the initialisation of the run. A critical issue is how to handle uncertainty within the data, which for some reason cannot be gathered. This is often achieved by performing some kind of sensitivity test, where the model is tested with different scenarios [Law, 2003].

Input data should first be studied in a broad sense to analyse which of them most influences the result. To reduce the amount of runs needed in cases where levels of different factors are to be evaluated it can be good to use design of experiments (DOE). DOE also saves computer power, even if this is of minor importance these days.

2.3.7 Production Runs and Analysis

Production runs, and their subsequent analysis, are used to estimate measures of performance for scenarios that are being simulated.

When analysing the output from a system there are two different types of systems to consider: [Sadowski, 1993]

- *Terminating systems*; These are systems that have a clear start time, and a clear end time for the operations. For these types of systems, it is required to establish the sample size and the simulation length.
- *Non terminating systems*; These are systems with no clear start time, nor a clear end time for the operations. For this type of system it is first necessary to bring the system into a steady state before the model generated data is used in an analysis. All the data generated by the model before it reaches this steady state should be disposed of. When this steady state is reached it is necessary to decide the length of the single replication to be run.

All scenarios should be conducted as they were described in the experimental design.

2.3.8 More Runs

Based on the analysis of the runs that have been completed, the simulation analyst determines if additional runs are needed, and if any additional scenarios need to be simulated. Multiple simulation with different random seeds are often preferred in order to establish a more general replication of the real system. The random seeds may not change the simulation outcome widely from time to time. Therefore, several runs with different seeds are necessary and the outcome from the simulation must be an average from the total number of runs.

2.3.9 Documentation

Documentation is necessary for a variety of reasons. If the simulation model is to be reused by the same or a different analyst, it may be necessary to understand how the simulation model operates. This will enhance confidence in the simulation model enabling a receiver to make decisions based on the analysis [Nordgren, 1995]. Also, if the model is to be modified, this can be greatly facilitated by adequate documentation. Proper documentation requires a minimum set of written documents [Banks, 1996] [Jägstam, 2004].

- Project specification

- Model documentation
- User documentation
- Final report

Preferably there are a number of other documents as well, all supporting the system in some way and all of them written continuously during the whole modelling process

2.3.10 Implementation

The successful result from the study is implemented in this last step. If the receiver of the simulation model has been involved throughout the study period, and the simulation analyst has followed all of the steps, then the likelihood for success in the implementation process is increased. Implementing the simulation model in the real world comprises many aspects. The implementation can be from changing a few parameters on a machine to implementing a new factory or a line. The implementation should be uneventful if the simulation was conducted properly and all the required personnel know what to expect.

2.4 Simulation Data

Simulation input data usually takes considerable time to collect which is a major problem. The simulation input data problem consists of a number of different problems: [Banks, 1996] [Law and Kelton, 1991]

- Availability
- Syntax and semantics,
- Information model,
- Dependencies, Autocorrelation, and Inhomogeneities,
- Information content
- Input data analysis.

The availability of input data is the main problem. Many companies seem to plan and control production with simple ad hoc methods, which makes correct data dispensable.

Others have the data, but it is well hidden in their information systems. The same data can also be in several information systems, but with inconsistent values. Yet another problem is the dependencies hidden in the data over several phases in a product or production system life cycle. It is difficult to have a complete map over all data that is used throughout an entire life cycle and different data is used in different phases, but the data has dependencies within itself, see Figure 9.

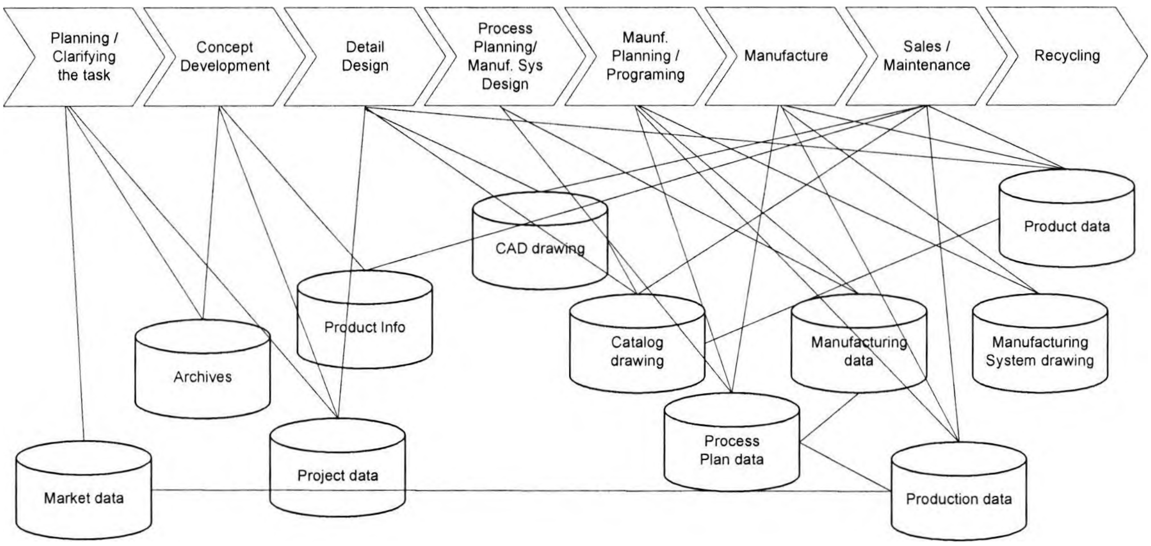

Figure 9 Different data sources for simualtion [Bernard, 2000]

In reality the steps in a simulation study are not discrete and separated in time [Trybula, 1994]. Instead they overlap. One of the major difficulties is the initial activities. The problem definition, problem analysis and data gathering become mixed together. After a short time, the pressure of delivering results or showing that something is being done causes the modeller to start gathering data, since the modelling and data gathering phase overlap, the scope of the effort constantly changes. This causes the model building to change direction, which extends the time of completion. This results in less time for verification, validation, experimentation and analysis and finally in a project that is late, incomplete and over budget.

To avoid the above mentioned problem Trybula proposes a philosophy that is based on developing reasonable estimates of the possible data values and to continue on with the normal model development adding real data as it becomes available. If data is not available, the study can aid in analysing possible output characteristics based on ranges of control variables. This structured approach provides a basis of minimising delays in the model building process due to lack of data.

Simulation tools rely on embedded models, i.e. data and logic are embedded in the model. Simulations are in general generated off-line with limited direct connections to the actual data. Instead, input data is gathered and analysed outside the simulation environment. This is stated to be the primary reason for models not being reused after the initial design usage of models. [Drake and Smith, 1996] [Peters et. al., 1996]

Seven factors that can lead to longer data collection time were presented by Liyanage and Perera. The presented factors are based on a literature review and a questionnaire survey conducted at the 1997 WSC. The list below highlights many of the problems mentioned above. The seven pitfalls were ranked in the survey and they are listed in the order in which they are considered to be influential. [Liyanage and Perera, 1998]

(1 — most influential, 7 — least influential).

1. Poor data availability.
2. High level of model details. It is stated that a higher level of detail does not necessarily lead to higher accuracy but to longer data collection time.
3. Difficulty in identifying available data sources. When data is available it is often located in different sources and in some cases the same data is in several sources. In the latter case the data might differ due to poor integration.
4. Complexity of the system under investigation. When the system is too complex, the simulation modeller tends to identify and collect data in an ad hoc manner.
5. A lack of clear objectives leads to indecision with regard to detail levels of the simulation model.
6. Limited facilities in simulation software to organise and manipulate input data.
7. Incorrect problem definitions, which cause the modeller to identify, collect and analyse invalid data.

As indicated by the survey the three most important and influential pitfalls with regard to simulation is input data management and the allocation and extraction of the right information from available information systems. This is one of the motivations for the work presented in this thesis.

2.5 Integration and Virtual Manufacturing

A central issue in controlling VM is finding a way of managing the complexity of the manufacturing system. The complexity arises from the number of processes that need to be controlled and coordinated, the number of products and variants to be managed, limited resources that must be considered, the various orders to be executed and several megabyte of data to be processed. [Vernadat, 1996] defines integration as:

> *Integration means putting together heterogeneous components to form a synergistic whole.*

And Vernadat also states that:

> *Essential conditions for integration seem to rely on the free but controlled flow of information and knowledge and the coordination of actions.*

The four types of integration that the industry has to choose between when making their data available for different types of processing have been listed [Vernadat, 1996]. First, two different levels of integration.

- **Loose integration**: Two systems are loosely integrated if they merely exchange information with one another with no guarantee that they will interpret this information the same way.
- **Full integration**: Two systems are fully integrated if and only if (i) the specificities of any one of the systems are known only to the system itself, (ii) the systems contribute to a common task, and (iii) the systems share the same definitions of each concept they exchange.

Furthermore, two other types of integration that are intended to coordinate the strategic, tactical and day-to-day decisions by implementing an efficient, timely information flow, and a structure, are also described. It allows the use of this information in an optimal way to control the physical flows. [Weston, 1993] [Vernadat, 1996]

- **Horizontal integration**: concerns physical and logical integration information, regardless of the organisational boundaries. This type of integration is usually dependent on the technology used and is realised at a given organisation level (e.g. plant level, cell level, station level). This type of integration mainly concerns the technological flow, i.e. the flow of materials and flow of technical documents.

- **Vertical integration**: concerns integration between the various management levels of the enterprise, i.e. decision-making integration, where management level defines the set of constraints for its lower management levels. This type of integration mainly concerns the decision flow, i.e. orders or objectives sent from the upper management level to the lower level and feedback information or status reports from the lower management level to the upper one.

However, the third type of integration from a VM perspective is mainly concerned with the activities within a given enterprise. Integration not only concerns the internal matters but can also involve different enterprises at different levels. Vernadat differentiates between two types of integration:

- **Intra-enterprise integration**: The integration of business processes internal to a given enterprise.

- **Inter-enterprise integration**: The integration of business processes of a given enterprise with business processes of other enterprises or sharing some parts of business processes by different cooperative enterprises. Inter-enterprise is the basis for the extended enterprise concept.

The fourth type of integration that has emerged over the last decades is within an enterprise. This is a complementary form of integration and it builds on the previous one. [Vernadat, 1996]

Figure 10 illustrates that physical system integration was first considered in the beginning of the 1970s and continued in the 1980s. This work predominantly focused on low level standardisation such as the seven-layer OSI-ISO definition. Physical system integration can only provide a limited level of integration. To increase this level, application integration had to be considered. The development of application integration began in the mid 1980s and is still very active, e.g. considering STEP for the exchange of common shared data in distributed environments. Again, only a certain level of integration can be achieved within the enterprise using this form of integration. To achieve full integration i.e. business integration one must enter the knowledge level of the enterprise.

- **Physical system integration**: concerns systems communication i.e. interconnection and data exchange by using computer networks.
- **Application integration**: concerns interoperability of applications on heterogeneous platforms as well as access to common shared data by various applications.
- **Business integration**: concerns integration at the enterprise level by coordinating business processes.

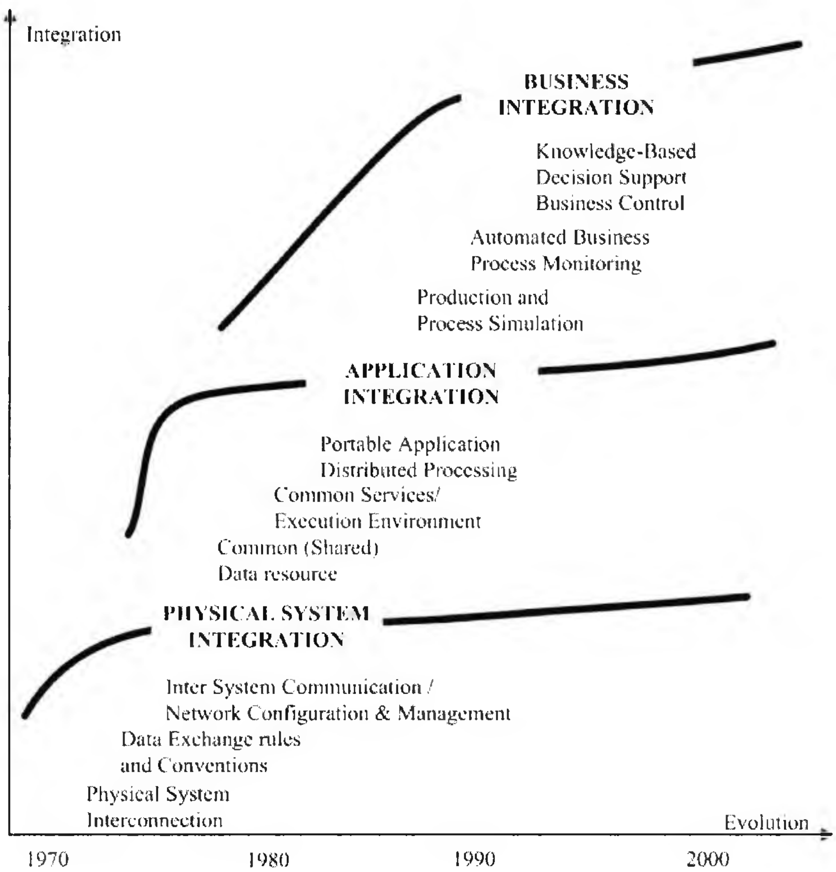

Figure 10 Integration levels [Vernadat, 1996]

2.6 Summary on Virtual Manufacturing

Much research has been conducted to support the development of manufacturing systems. A computerised support for manufacturing systems which virtual manufacturing constitutes has gradually emerged during the past decades. The research within virtual manufacturing has fractured into several directions where the main purpose of virtual manufacturing is described as a method of creating an integrated computer based model which represents the physical and logical schema of the behaviour of a real manufacturing system.

A number of different models with accompanying tools can be found in computer simulation [Banks et. al., 1996] [Gogg and Mott, 1993] [Law and Kelton, 1991]. Discrete event simulation is one of the tools which is also used in this research.

It can be concluded from literature that by following a methodology during the simulation modelling and analysis the result can be improved. One of the most acknowledged methodologies of performing DES is the one provided by Banks [Banks et. al., 1996]. As described by the methodology data collection and model conceptualisation is one of the key issues in cutting costs during modelling. It is therefore essential to focus extra on data management. The integration of information sources has been of special focus during last decade, aiming to overcome much of the manual cut and paste work which otherwise has to be done.

3 Information Management and Standards

It has always been important to manage information properly. However it is not as easy as it may appear. An increasing amount of information will make it more difficult to manage without proper support and understanding. In order to succeed, a thorough understanding of information and help from adequate standards is a necessity.

3.1 Data, Information and Knowledge

Data is defined as signs or symbols that, depending on which rules they are based on, can represent information when it is interpreted [Schenck and Wilson, 1994]. A similar definition is made by [Holmer et al, 1990] who mean that data is more like discrete facts of the real world it represents. However, the tenor of the definitions is that data is the raw material, the building blocks of the thing we call information.

Schenck and Wilson argue that data is symbols which represent information for processing purposes and that it is based on implicit or explicit interpretation rules. Thus it can clearly be said that data itself has little or no meaning and says nothing about its importance or relevance. As the building block data does not represent a concept. It is not until the data is fitted to a concept that it brings a meaning. When an association between the data and the concept is made the data is understood and becomes information.

What if the data is associated with more than one concept or with a different concept than for which it was initially created for? Does the data still become information when it is read into another concept? For simplification, in this thesis, information is considered to be data interpreted by its original meaning. However, the problem is discussed in more detail in section 3.1.1.

Further, Schenck and Wilson [Schenck and Wilson, 1994] argue that information is knowledge, whereas Hicks [Hicks et al, 2002] and Holmer [Holmer et al, 1990] mean that knowledge is the result of a cognitive process. Hicks et al. even differentiate two types of information which they claim are the fundament on which the cognitive process relies. They are:

- *Formal information* is an element of information that provides a specific context and measure as it also provides a structure so that people exposed to it may infer the same knowledge from it e.g. education. Content and order are prescribed. For the purpose of communication, formal information can be divided into three sub-categories.

  1. *Textual* (structured) may consist of numerical, alphabetical or symbolic information or combinations thereof. The medium for communication could be paper based or electronic. The information is expressed in a format that is an accepted language or agreed symbols.

  2. *Pictorial* (structured) information is considered to be any visual image conforming to accepted standards. Examples are diagrams, two and three dimensional drawings and flowcharts.

  3. *Verbal* (explanative) information is expressed in a logical and structural manner. It is typically used in a professional situation when something is described between two individuals.

- *Informal information* mainly consists of information that is personal or that is developed through interactions between two or more individuals. In this type of information the subject and predicate may be unclear and the information may change dramatically as the content is continuously altered and added to. Further, it is those sudden changes that stimulate and develop the creative and decision making processes. The information can be divided in to five sub categories:

  1. *Memory* is the information considered to be within a person. This type of information may be generated by past experience, both formal and informal. The content and relevance may also be unclear and ill-defined.

  2. *Textual* (Unstructured) can be expressed in a personalised notation or in an accepted language. The information does not necessarily follow any logical scheme and it may also comprise incomplete sets of information that are merely pointers to true and meaningful information.

3. *Pictorial* (Unstructured) is similar to textual unstructured information because it may not follow any logical order or conform to any standards. Typical information of this kind is sketches and outlines to diagrams.

4. *Verbal* (Conversational) is a dynamic process. A discussion may include information that is not clearly defined and the topic can change over time. The information sets are altered, added to or removed as the discussion progresses.

5. *Expression* includes both physical expression and intonation of the voice when something is said. Expressions are used to show approval, indifference and dislike and are powerful contributors when passing information. They are based on information within a person and may not be truly quantifiable.

Another definition of knowledge is the one provided by [Holmer et. al., 1990] which states that knowledge forms the sum of experience and information a person has collected and more or less consciously structured for his/her needs. Further, the use of knowledge depends on the situation where it has been collected. Figure 11 demonstrates the correlation between different types of knowledge and when they can be used for solving other future situations that may demand prior knowledge.

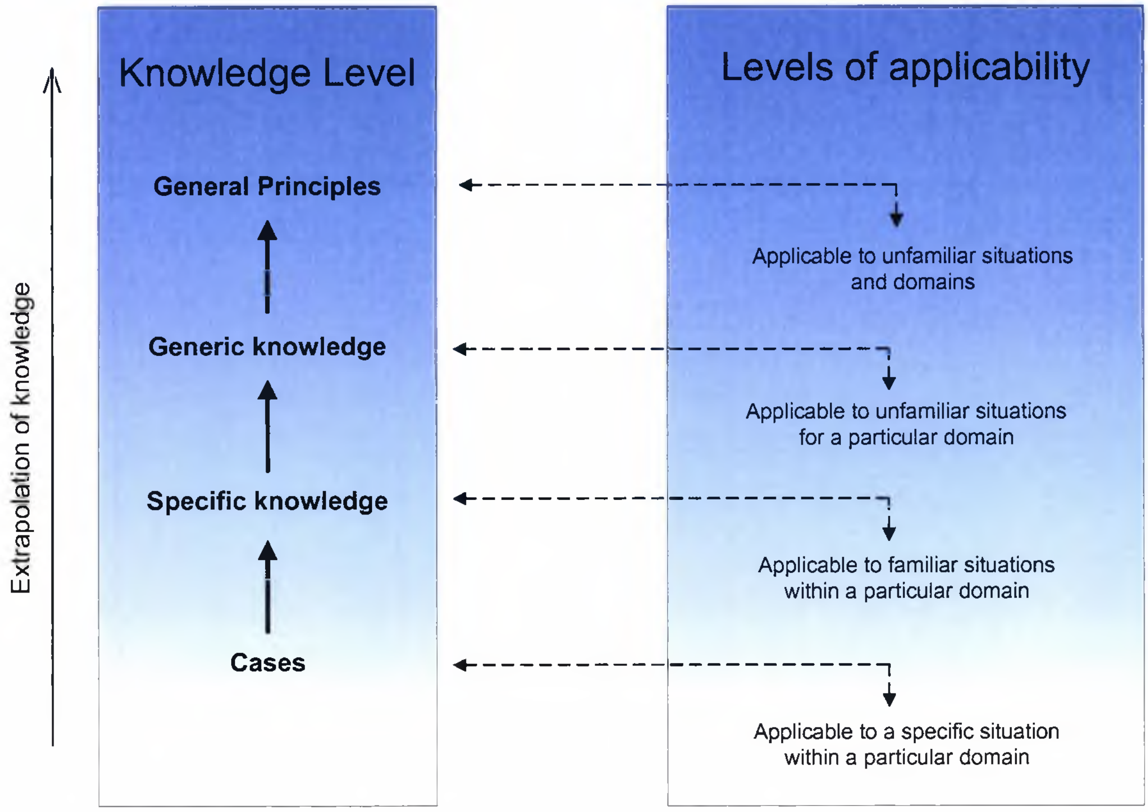

Figure 11 Knowledge levels and states of applicability [Hicks et al, 2002]

It is the ability to use information to make the right decisions on a day to day basis that is called competence by professionals. Further, competence is not only knowledge and the ability to make decisions based on it but also the motivation to effect change in an organisation. This is described by Bergman and Klevsjö [Bergman and Klevsjö, 1998] as the possession of the right skills and knowledge to perform a service, Figure 12. Without knowledge and competence, good decisions and actions cannot be made.

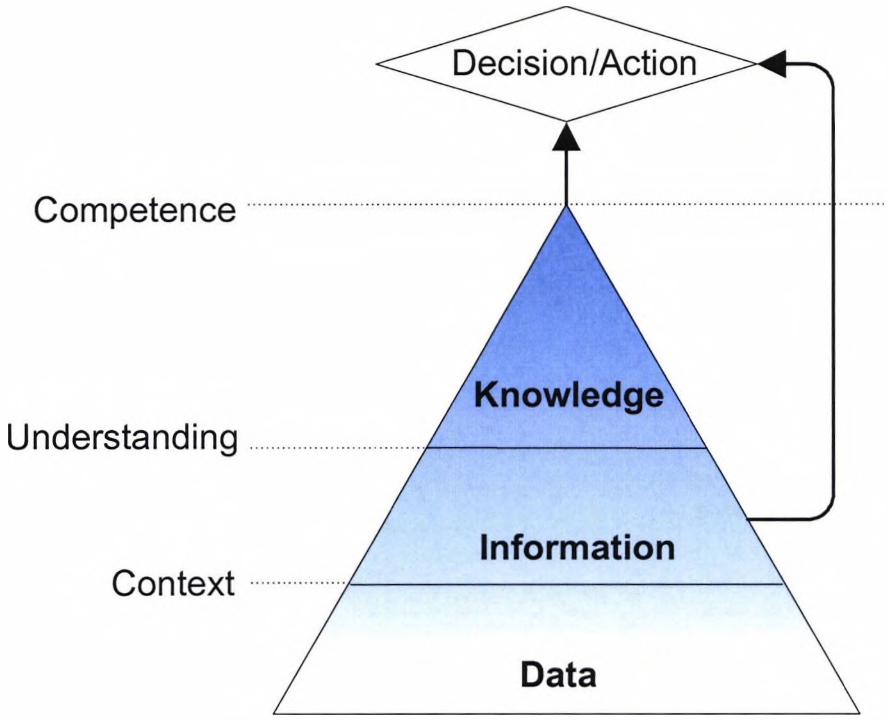

Figure 12 The relationship between data, information and knowledge [Bergman and Klevsjö, 1998]

3.1.1 Information Modelling

For most companies it is important that its employees create the same knowledge from the same information based on common data. However, it may be impossible to obtain the same knowledge from an identical information source because of the individual's interpretation of it. Unity of knowledge and information understanding can only be assured if individuals have access to a formal description of how to interpret the data and information. It means that the data and information must be provided with a specific context, structure and rules of interpretation which an information model would typically supply.

An information model illustrates how data should be interpreted to become information. This is also congruent with the definitions provided above. A rather motivated question of what an information model means now presents itself. Information models help interpret the data correctly where implicit or explicit rules of interpretation are needed. An information model provides the explicit interpretation rules of the data in a formal way. There are always two rules that need to be fulfilled for an information model:

- *Purpose*, what is the aim of the model and what is it trying to communicate.
- *Viewpoint*, who constructed the model and what is that person's intention with the model.

Those two rules distinguish the user from the usage of the model.

Schenck and Wilson use a formal approach to describe the interpretation of data to form information. If **I** and **D** stand for Information respectively Data, then **R** is the set of explicit interpretation rules that help produce information from data. Thus formally

$$\mathbf{R(D) \rightarrow I}$$

means that data is transformed into information by applying the interpretation rules. The opposite is also possible, by applying the inverse rule $\mathbf{R_{-1}}$ to information, data is produced. This is formally expressed as

$$\mathbf{R_{-1}(I) \rightarrow D}$$

If **R** is invariant then information can be exchanged without loss. That is formally described as

$$\mathbf{R(R_{-1}(I)) \rightarrow I}$$

This type of reasoning is important to explain the purpose of information models and information modelling. However, in real life the optimal situation is seldom a norm, therefore to simulate the physical conditions the reasoning has to introduce interpretation rules for both the sender (indexed by **s**) and receiver (indexed by **r**) into the description. The formal description is

$$\mathbf{R_r(R_{-1s}(I_s)) \rightarrow I_r}$$

This description means that information from the sender is transformed into data, passed to the receiver and transformed back to information by the receiver of the data. However, this is only successful ($\mathbf{I_s{=}I_r}$) if $\mathbf{R_{-1s}}$ is equal to the inverse of $\mathbf{R_r}$ thus, $\mathbf{R_s{=}R_r}$. Thus information can only be exchanged fully and with a preserved meaning if the rules of interpretation are the same for both the sender and receiver. In relation to information modelling the model provides the explicit rules of interpretation in order to ensure that the intended meaning of the information is always preserved.

Further, Schenck and Wilson distinguish between two types of information models; conceptual models and concrete information models. On the one hand the conceptual model is independent of any instantiation model. This means that it has not been constrained by any limitations that might have been imposed in an instantiation model. On the other hand the concrete information model is developed with a particular instantiation method in mind and may include more refined solutions that do not need to be a general solution.

3.2 Data Management Systems

There is a rough general agreement on the basic functions a Data Management (DM) system should provide [Katz, 1990] [van den Hamer and Lepoeter, 1996], even if terminology and details vary considerably.

A list of the basic functions is given below. The term 'object' in the list refers to an entity, such as a product or a document, managed by a DM system.

- *Secure storage of documents and other objects in a database with controlled access.* In many companies the first motivation for considering DM comes from people frustrated by being unable to find the documents they are looking for. These people may only know that the needed documents are located somewhere in a network of file servers. There may even be many copies of a document at different locations, and in the worst case scenario two people can make conflicting changes in two copies of the same document.
- *Mechanism for associating objects with attributes.* The properties of documents and other objects are described by means of attributes. The attributes provide necessary information about an object, and they can also be used for finding objects.

- *Management of the temporal evolution of an object through sequential revisions.* Many DM systems were originally built for design and development environments. In these environments the users typically spend more time modifying existing designs than creating completely new ones. The evolution of drawings and other design objects is usually captured in the form of successive revisions.
- *Management of alternative variants of an object.* Many products and documents have alternative variants. For example, a user's guide for a particular product can be available in different languages.
- *Management of the inspection and release procedures associated with the objects.* Documents and other objects with engineering data must typically be checked and approved by more or less elaborate procedures before the objects are released for general use.
- *Management of the recursive division of an object into smaller components.* Almost any product has a hierarchical breakdown structure. This divides the product into components, which are then further divided into smaller subcomponents, etc.
- *Management of changes that affect multiple related objects.* One of the primary functions of a DM system is to support change management. In their basic form the revisions and variants represent the changes of separate objects. Nevertheless, it is often also necessary to view a set of related objects as a single unit with respect to change management.
- *Management of multiple views of an object.* A DM system should make it possible to have different views of an object. For example, a product can be divided into components in more than one way.
- *Management of multiple document representations.* A DM system should also make it possible to store a document in multiple different presentations. For example, a drawing created with a CAD tool can be available both in the native file format of the tool and in a neutral file format for viewing and printing.

- *Viewing tools.* In addition to simply displaying the read-only representations, some viewing tools allow users to insert textual and graphical annotations on top of the documents without modifying the original data.
- *Tool integration.* From an ordinary user's point of view, the usability of a DM system depends very much on how well the system is integrated with other tools that he or she needs in his or her daily work.
- *Component and Supplier Management.* The management of standard components bought by a company from external suppliers is a rapidly growing field within DM.

The functions are not independent of each other. It is, for example, difficult to discuss sequential revisions without the idea that an object under development is released for use after it becomes "ready". Many of the items on the above list can be seen as part of *configuration management*, which is an engineering discipline and a process for maintaining the integrity of products while they evolve through development and production cycles [Buckley 1996].

Further, there are a large number of systems on the market that supports the product realisation activities and are starting to cover the production area as well. Companies have often tried some kind of DM system to support different activities in different processes. In this thesis only three DM systems are presented, which are also considered to be closest to the scope of the work. However, for DM systems outside the manufacturing and simulation area e.g. economic conditions may be different than stated here.

3.2.1 PDM

Product data management (PDM) systems are very product centric and manage documents and product structures throughout the whole product design. Designers use these systems to organise the work and to break down the product design into manageable parts which are then distributed among the designers. PDM systems also support workflows, which can be used for planning and executing projects [van der Aalst and van Hee, 2002]. Often some of the business processes mapped in the workflow module will ensure that the process is executed as intended. There can be various types and complexity of PDM systems, from simple document handling systems to ones capable of handling all the information about a product and its configuration, structures, variants, etc. As all DM systems have access control, this automatically gives the user access to all the information that the users' role should have. Thus the system is disconnected from individuals and instead incorporated in the organisation in a natural way. [Peltonen, 2000]

3.2.2 MDM

Manufacturing Data Management (MDM) systems are a new generation of data management systems which are more focused on the production aspects than older more traditional PDM technology. Further, MDM systems originate from the basic ideas incorporated in the concept of PDM and can at a very intermediate level seem just as another PDM system [Hicks et. al., 2002]. However, the major difference between the MDM and PDM systems lies in the ability to handle process information that is tied to the product. The MDM systems are trying to produce a better representation and support of the whole development process during all phases of a product, from a production point of view [Johansson, 2001]. This includes various aspects of the product and production systems from the concept phase to the runtime and maintenance of the production system as well.

3.2.3 ERP

Product data outside a PDM system must be managed with other systems. Instead of using a large number of separate systems, it is becoming increasingly popular to manage this data in large integrated Enterprise Resource Planning (ERP) systems, such as SAP R/3, Baan IV and Oracle Applications. Since the goal of ERP systems is to comprehensively manage almost all data within a company, the functions typically included in a PDM system should logically be found as part of ERP systems. In fact, ERP systems are beginning to include PDM functions, sometimes as separate modules. Nevertheless, at least at the moment, the PDM functionality in ERP systems is rather limited, and an ERP system is seldom a substitute for a PDM system [Peltonen, 2000]. However, there is obviously overlap between these two kinds of systems and there is a strong need to share common data.

3.3 Information Standards

DM systems are used in a large number of concepts for which there are no universally accepted standards. For example, practically all DM systems have some notion of versions, but there is no agreement on the meaning of the terms ‘version’, ‘revision’, ‘variant’, etc. Efforts are being made to develop standard support for DM systems in order to facilitate data sharing.

3.3.1 STEP ISO 10303

One effort is the International Standard for the Exchange of Product Data, officially known as ISO 10303 and generally referred to as the STEP standard [Kemmerer, 1999]. The scope of the standard is very large due to the fact that the goal was to develop a standard for the representation of all data related to products. The first parts of the standard were accepted in 1994, and new parts are continuously being developed and accepted as part of the standard. In spite of the many years of hard work to develop STEP, the effect of STEP on the DM field is still only marginal.

STEP includes a number of parts which are divided into several classes, the most important of which are described here. A very large portion of STEP deals with the representation of geometric data. At the moment, this is probably the most successful part of STEP. If STEP is to serve as a universal basis for product modelling tools of the future, it must clearly support generic product structures. [Johansson, 2001]

3.3.2 Express Language

STEP defines a number of data models for various aspects of product data. All data models are defined by means of a schema definition language called EXPRESS [ISO/TC184/SC4, 1994a]. EXPRESS can be regarded as a semantic database modelling language [Hull and King, 1987]. This means that EXPRESS can be used for defining a database schema that consists of object types, their attributes, relations between objects and validity constraints.

It is important to understand that EXPRESS only describes the structure of data, not any behaviour. Given a population of objects (i.e., a database instance) and an EXPRESS schema, one can say whether the object population is valid with respect to the schema. EXPRESS is not a programming language and does not include any mechanism for defining operations or methods for the objects described in the schema. Just a brief look at EXPRESS may be misleading because the language includes procedures and functions, which can contain familiar programming constructions, such as variables, conditional statements and loop statements. Nevertheless, these constructions can only be used in validity constraints and definitions of derived attributes. A validity constraint can thus be an arbitrarily complex algorithm written in a "programming language" but all the same the algorithm only defines a test for determining whether an object or a population of objects is valid.

EXPRESS-G is a graphical notation for schemas defined in EXPRESS. An EXPRESS-G diagram can only show entities and their attributes (including references to other entities) but not any rules. However, EXPRESS-G has the possibility of indicating where rules are applied. An EXPRESS-G diagram provides an incomplete but comprehensible overview of a schema, which must be described in detail with the textual EXPRESS language.

3.3.3 External Representation of EXPRESS Data

The original goal of STEP is to make it possible to exchange product data between organisations. STEP Part 21, called 'Clear text encoding of the exchange structure' therefore specifies how a set of instances of entities defined in an EXPRESS schema are represented as a textual physical file [ISO/TC184/SC4, 2002]. As a very simple example, consider the following schema:

```
SCHEMA residences;
ENTITY person;
name : STRING;
hometown : town;
END_ENTITY;
ENTITY town;
name : STRING;
country : STRING;
INVERSE
citizens : SET OF person FOR hometown;
END_ENTITY;
```

A physical file with a couple of instances of these entities could look as follows:

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION(...);
FILE_NAME(...);
FILE_SCHEMA(('residences'));
ENDSEC;
DATA;
#1 = TOWN('Reading', 'United Kingdom');
#2 = TOWN('Washington, D.C.', 'USA');
#3 = PERSON('Sahra', #1);
#4 = PERSON('Bill', #2);
ENDSEC;
END-ISO-10303-21;
```

Inverse (and derived) attributes do not appear in the physical file because their values are automatically determined from the values of the explicit attributes. In addition to a text file, EXPRESS data can be stored in a repository, which can, for example, be implemented as a file or a database. The contents of a repository are accessed by means of operations defined as a Standard Data Access Interface (SDAI) in STEP Part 22. The SDAI is thus an Application Programmers' Interface (API) to an EXPRESS repository. Part 22 is independent of any programming language. Further parts define language bindings, which describes how the operations of the SDAI are used from an application written in different programming languages such as C, C++ and Java.

Part 28 is an ISO Technical Specification, which means that it does not have the same status as a full standard and is intended to be revised and updated as the technology changes. At the STEP meeting in Portugal in February 2001, the ballot comments for Part 28 were discussed. A revision of Part 28 is currently being made to provide a clearer justification for the two early bindings and improved integration between them. Part 28 enables the output of STEP information in XML format, which can be used to in provide support for internet usage of STEP models and also new areas not yet developed.

3.3.4 Integrated Resources

Many application protocols must represent similar data. For example, almost all application protocols include some kind of geometric data. The application protocols are built from integrated resources, which are divided into generic integrated resources and application integrated resources. As the names suggest, the former ones are intended for all kinds of applications while the latter have been developed for a range of similar applications.

3.3.5 Application Protocols

The actual standardised product data models are defined as STEP application protocols (AP). Basically the definition of an application protocol consists of two parts. The first part is an application reference model (ARM), which defines the necessary concepts for the application domain to be covered by the application protocol. The application reference model is in principle written with regard to its eventual implementation. The second part of an application protocol is an application interpreted model (AIM), which shows how the application reference model is implemented with the integrated resources. An application interpreted model provides an interpretation to entities defined in the integrated resources. An application interpreted model can also define entities as subtypes of entities in the integrated resources.

What STEP basically standardises are the application protocols. In order for two applications to exchange STEP product data, the data must be exchanged with some application protocol "understood" by both applications. Sometimes a software tool, such as a PDM system, is advertised as "STEP compliant", "STEP compatible" or something similar. These statements make little sense unless one specifies that the tool can manipulate data according to a particular application protocol. In extreme cases, the fact that the data model of a system has been described in EXPRESS may be used as a justification for touting the system as "STEP based". There are problems with application protocols. The most immediate observation that can be made from an application protocol standard, such as AP203 for configuration controlled design [ISO/TC184/SC4, 1994b], is the almost incomprehensible format of the standard. For example, the bulk of AP203 consists of the specification of the application interpreted model in the form of over 200 pages of mapping tables and rules. Most concepts are presented in alphabetical order, making it almost impossible to form an overall picture of the standard.

Application protocols are independent of each other. All application protocols are based on the same integrated resources but the application level concepts that are built on top of the integrated resources can vary in different application protocols. This causes problems if different application protocols should be used together [Teeuw et. al., 1996].

STEP was originally designed for data exchange. The problems with integrating different application protocols become especially pressing when STEP is used as a basis for sharing and archiving product and process data within an organisation.

3.3.6 Open PDM Schema

An initiative from the STEP community is trying to consolidate information and enable applications to work together more easily. Since the information that can be represented by the standard is quite extensive in its full format, an alternative of working with the information is available. By utilising a meta-structure of the represented data, support for DM systems can be achieved, with much faster and easier management.

The open PDM schema is built based on four application protocols, Configuration-controlled design (AP 203), Electro technical design and installation (AP 212), Core data for automotive and mechanical design processes (AP 214) and Technical data packaging: core information & exchange (AP 232), see Figure 13.

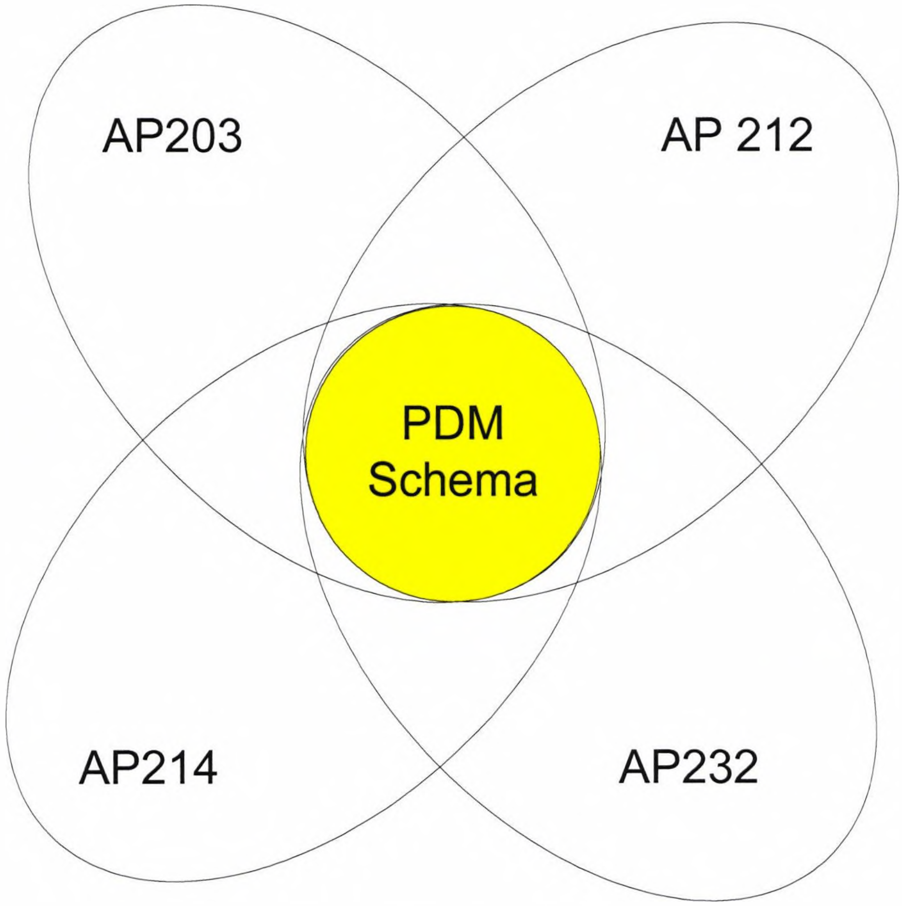

Figure 13 Open PDM Schema [Oh et. al., 2001]

The structure enables an exchange of references between the application protocols which makes sure that the information is kept in its original application protocol. Only non geometrical information is exchanged. [Oh et. al., 2001]

3.4 Interfacing with Other Standards and Techniques

ISO TC184 subcommittees prepare standards relating to industrial data. Subcommittee 5, Architecture, Communications, and Integrating Frameworks, has a working group called WG1 that deals with modelling and architecture. WG1 has prepared standards of enterprise models such as, ISO 14258, and enterprise-reference architectures, ISO 15704. These are high-level standards, focusing on enterprise-level concepts. ISO 10303 usually focuses at the process level where product information is exchanged among engineering and manufacturing applications. When the enterprise itself is a product and a project of some organisation, then the differences between levels are erased. An example of where an enterprise may be a product or project is an architectural, engineering, and construction company, that designs, builds, operates, or disassembles enterprises. The enterprise (product) must be represented over its entire lifecycle, the scope of ISO 15704. The enterprise model then becomes a product model. In this case, there is considerable opportunity for a STEP AP project to consider shared tools for product representation, for example, using EXPRESS to represent an enterprise model. Some of the enterprise-reference architecture components of ISO 15704 may also prove useful (i.e., reusable enterprise-reference models, enterprise-engineering tools, and applicable enterprise-engineering methodologies). SC4 and SC5/WG1 have explored areas where they can use each other's technology and standards. The most immediate application is the set of architectural, engineering, and construction application protocols: 10303-221, - 225, -227, and -230. Since ships and buildings are similar, the shipbuilding APs also could apply: 10303-215 through -218, and -226. Other areas needing coordination between SC4 and SC5 are ISO 13584 (Parts Libraries), and ISO 15531 (MANDATE) [Kemmerer, 1999]. WG1 is planning new standardisation work at lower enterprise levels and SC4 and SC5/WG1 anticipate a by-product of that work will point to further coordination opportunities.

3.4.1 Common Object Request Broker Architecture

CORBA is the Common Object Request Broker Architecture developed by the Object Management Group (OMG) [OMG, 1997], a consortium of over 600 members including many software vendors. CORBA defines an integration technology that allows diverse object-oriented applications to exchange data in a 'conversational' mode, independent of specific platforms and object implementation techniques.

Every CORBA transaction starts with a client request for information and ends with a server response. In subsequent transactions, the roles of the client and server applications may be reversed. Each information request is routed via an Object Request Broker (ORB) that identifies the appropriate server to provide the required information. The ORB maintains a directory of servers and their services, together with details of their interfaces of the integrated system, both in the client and the server role if appropriate.

In CORBA an object is "an identifiable encapsulated entity that provides one or more services that can be requested by a client." Thus, the application programs in an integrated system are regarded as objects. The IDL language permits interfaces of such objects to be defined for CORBA purposes, independently of the actual implementation of the object. Provision of one form of interoperability between STEP and CORBA is under way through an IDL binding [ISO/TC184/SC4, 1998], which is being developed for ISO 10303-22. This will allow a STEP model in a database to be treated as a server in a CORBA implementation.

The types of "objects" dealt with by CORBA exist at the level of entire models or files from the STEP point of view, rather than at the level of individual entities within models. Thus, the primary relationship between STEP and CORBA will be in the area of PDM. Resolution of incompatibilities between the STEP and PDM approaches to handling product configuration management data may have some influence on the future development of the STEP architecture. SC4 and OMG standard developers are actively working toward harmonising the way PDM data is handled.

3.4.2 Internet and Intranet

The Internet and the Intranet are playing an increasingly larger role in the daily lives of the average European, as well as the average engineer. The Internet is being widely used for access to data and information on a global basis.

Many commercial companies are using the Internet for advertising their products and providing catalogue information of their products. This media provides standard part suppliers, as well as custom design businesses, with little or no additional cost an opportunity to advertise and make their product available to a broader market.

Intranet use within companies and organisations is broadening because of the ready access to data in a format that is compatible with Hypertext Markup Language (HTML) browsers.

It provides company access to such information as standard part data, data or drawing viewers, release information, and status in a guaranteed secure fashion.

The marriage of STEP and the Internet offers some exciting prospects for the future of STEP. STEP thrives in a networked implementation environment. Unfortunately, the early STEP implementations were specified before the Internet explosion. The 10303-21 text file exchange, with its dependency on special-purpose parsers, has not easily found a home on the Internet. The SDAI is a single user data access interface for STEP-based applications; it is not designed to support networked applications. OMG's CORBA may provide some of the tools to bring STEP to the Internet. For example, CORBA promises location transparency for STEP models across an ORB-enabled Internet.

With CORBA it will be possible to find a STEP model without knowing its precise location; however, the CORBA distributed object paradigm is not designed to support the STEP requirement for moving data objects from one location to another. Thus, while CORBA may help to bring STEP to the Internet, it is not sufficient in itself. The future of STEP on the Internet depends on the ability of defining an effective integration of STEP and Java. A project within ISO TC184/SC4 is examining this challenge. The goal of the project is to make EXPRESS-based [ISO/TC184/SC4, 1994a] data objects as accessible on the Internet as HTML objects. The Java pass-by-value paradigm is the enabler that will make this possible. A Java-STEP Internet will rely only on proven Internet technologies: Hypertext Transfer Protocol, HTML, Java, and Java Object Serialisation.

With Uniform Resource Locators as persistent identifiers for STEP and EXPRESS-based data objects, the Internet becomes a worldwide repository for shared product data. Moreover, the Internet also becomes the clearing house for libraries of STEP EXPRESS classes. With Java's "write once, run anywhere" potential, these classes can be downloaded from the Internet and executed anywhere. Java is a key to the popularisation of STEP. There are more than 400,000 practicing Java programmers who are producing new Internet applications at an amazing rate [Kemmerer, 1999]. The Java programming environment promises to provide even greater programming productivity.

PDES, Inc., in collaboration with NIST, proved through the United States Air Force PAS-C Program demonstrations that data access through HTML viewers is viable and usable through fairly inexpensive products (e.g., Netscape).

The data structures within 10303-232 for top down breakdown (i.e., part, document, or mixed) and file information for the top down breakdown is defined for that capability to be exercised. 10303-232 also provides the data structures which catalogue information for products with part family and part classification information. A populated 10303-232 data file provides a capability, when the instantiated data is converted to HTML format, to provide users relatively inexpensive access to data without expensive CAx applications. HTML does not provide data structures for a data model, but an ability to relate different 'existing' data together.

The extensible Markup Language (XML) can enable product data exchange as an alternative to the existing ISO 10303-21 as an encoding of the STEP schema instance. Although XML will probably be most suitable for the exchange of product data that is not geometry-intensive (e.g., change orders) and where exchange files are not overly large, it enables WEB-based distributed PDE implementations. XML is a standard being developed under the support of the World Wide Web Consortium (W3C). It is a format for structured data interchange over the Internet, and most Internet browser vendors supports XML in some way. XML is important to industry and to STEP because it [Kemmerer ,1999]:

- Supports data exchange between heterogeneous systems.
- Data sharing between manufacturing applications and common business software tools.
- Facilitates electronic commerce.
- Reduces start up costs.
- Enables interoperability between different transaction processing systems.
- Enables seamless integration between Internet and desktop.

- Provides an on-ramp to the Internet for data represented using standards developed prior to the ascendancy of the Web, such as EDI or STEP.

Why XML instead of HTML? Well, XML is extensible (content providers can develop their own tag sets); HTML is not. XML documents must be either valid with respect to a document type definition, or they must be well formed; HTML documents may contain tagging errors. XML is designed for representing structure; HTML is designed mainly for presentation. XML documents are intended for interpretation by applications (after being processed by a parser); HTML documents are intended to be read by humans.

3.4.3 Electronic Data Interchange (EDI)

EDI is the exchange of business data between trading partners, as defined by the ANSI-Accredited Standards Committee (ASC) X12 standard in the U.S. or by the United Nations Electronic Data Interchange for Administration, Commerce and Transport United Nations (EDIFACT/UN) standard. Since it is often a requirement to associate product data with business data, it is clearly desirable for STEP to interoperate with EDI. A project performed for NIST and the U.S. CALS Office studied the requirements for EDI and STEP to work together. A combination of the two types of data can be considered the components of a 'technical data package'. Initially, the Department of Defence (US) technical data packages were examined to establish the scope of the required items of information and their interrelationships. Short, medium, and long-term strategies were defined for achieving interoperability. The short-term solution relied mainly on the capabilities of EDI, but work on the medium-term solution is already under way within ISO TC184/SC4 in developing 10303-232.

STEP and EDIFACT are viewed as complementary standards, addressing different applications in the field of electronic commerce. It is important to note that harmonisation between the two standards will not necessarily lead to any modification of any of the standards. In the long run, harmonised definitions of concepts, through the use of a single dictionary, are envisioned.

3.4.4 Object Linking and Embedding (OLE)

Object Linking and Embedding (OLE) is based upon the Component Object Model (COM) jointly developed by Microsoft and DEC. In effect, OLE provides a mechanism for constructing compound documents (in a generalised sense), regarded as objects, and COM is the associated means for communication among distributed objects [Microsoft, 2005]. The expectation is that it could become a de facto standard, at least for PC-based applications. CORBA, as earlier described, has been developed by the rest of the software industry to serve much the same purpose as COM, and various options exist for an associated compound document format. Product lifecycle application software CAD vendors are beginning to migrate to the use of PC platforms. If vendors uniformly continue this migration, it may become very important for ISO 10303 implementations to interoperate with OLE and COM. This interoperability is for the same reasons that interoperating with CORBA is a current requirement. Significantly, an extension to OLE, known as OLE for Design and Modelling, is currently under development. At present, this seems to be restricted to handling the spatial arrangements of graphical objects, but further extensions could bring OLE for Design and Modelling closer to the scope of STEP and increase the need for interoperability. Currently, however, there is no work in progress towards this end.

3.4.5 Java

Java [Gosling et. al., 2005] is a platform-independent, object-oriented programming language developed by Sun Microsystems. It allows a program to be written once and then run anywhere on the Internet. Interoperation of STEP with Java could involve, for example, the use of Java programs for visualising STEP models over the Internet. Originally the Java language specification was submitted to ISO/IEC JTC1 for publication as an international standard using the Publicly Available Specification process. Now, Sun has opted to pursue the technology's standardisation first through the European Computer Manufacturers' Association. Work has started within ISO TC184/SC4 on a Java binding to the STEP SDAI.

3.4.6 Mandate

The Manufacturing Management Data (MANDATE) standard (ISO 15531) is intended to cover standardised representations of manufacturing information other than product-related data. This includes such topics as manufacturing resources, materials flow, and managing manufacturing. When detailed MANDATE models are created in any of the relevant application areas, it is suggested that they will created, based upon ISO 10303 resources and be designed for interoperability with ISO 10303 application protocols. [ISO/TC184/SC4, 2005]

3.4.7 Standard Generalised Mark-up Language

SGML is the Standard Generalised Mark-up Language [ISO 8879, 1999], an ISO/IEC standard for computer-based documentation. It has already been suggested that a close connection should be created between this standard and STEP since much of the information in a product description consists of textual documents. The aim is to enable creation of structures in which SGML documents can be embedded in STEP files, with appropriate references between EXPRESS-based product information and the SGML documents, and vice versa. This will lead to a major and very desirable expansion of the STEP concept of the product representation. An added of effect of interoperability of SGML with STEP will be that the EXPRESS string, currently a non-computer-interpretable data type, will gain intelligence. Thus, in principle, SGML strings transmitted in a 10303-21 file could be subjected to a further level of interpretation after post processing of the file. It is significant that strings defined by any other ISO standard are also valid SGML strings, therefore these computer-interpretable strings could include (for example) code segments in programming languages such as C, or FORTRAN. The embedding of EXPRESS strings in 10303-21 files would also become valid, and applications for this capability have already been identified.

3.4.8 Virtual Reality Modelling Language

Virtual Reality Modeling Language [ISO/IEC, 1997] is sometimes mentioned as an 'alternative' to STEP. In fact, the two have very little in common. VRML was developed primarily for creating interactive 3D simulations on the World Wide Web. It provides purely graphical capabilities, and has no provision for representing non-shape-related engineering data. The language provides several Constructive Solid Geometry-type primitive shapes for visualisation, but shapes that are more complex must be represented by polyhedral approximations. VRML is in the public domain. It is being developed as an ISO/IEC JTC1 standard in collaboration with the VRML Consortium, and is based on the Open Inventor modelling format developed by Silicon Graphics Inc. There may be virtue in providing means for translating STEP shape models into VRML models for visualisation on the World Wide Web; however, this appears to be the only form of interoperability that is likely to be useful.

3.5 Other Initiatives within the Manufacturing Community

The idea behind the effort of developing enterprise integration frameworks and information sharing protocols has been that a large part of the business process reengineering projects are similar in their behaviour in every enterprise. Thus, they could be captured, standardised and reused again instead of developing them over and over again. One standardised generally accepted framework can support models, methodologies and a range of products and efficiently save both time and money. The ideas came from other engineering areas such as electrical and software engineering.

Over time several enterprise models have evolved and some of them have been more useful than others. This thesis will only briefly point out the characteristics in some of the more accepted ones, which will illustrate how the work presented is influenced by several of the reference architectures.

3.5.1 CIMOSA

The European Open System Architecture for CIM (CIMOSA) has been developed by the AMICE consortium. The goal of CIMOSA is to help companies to manage change and integrate their facilities and operations to face worldwide competition. That includes price, quality and delivery time. CIMOSA has been driving the idea of vendor independent manufacturing modules. The modules have to be described in terms of function, information, resources and organizational aspects. This would help the development towards a plug and play approach when designing an enterprise. CIMOSA has:

- **Architectural framework**, which comprises of three major components. (i) an integrated modelling framework, (ii) and integrating infrastructure and (iii) a CIM system life cycle.
- **Functional Entities**, in CIMOSA are all active resources which are able to execute basic functional operations of an activity and also play all roles in a process. This can be said to be the fundamental concept in CIMOSA which provides modularity and genericity.
- **Modelling Framework**, promotes descriptive modelling, therefore it models the enterprise with a non overlapping set of building blocks to cover various aspects of the enterprise. The CIMOSA modelling framework is based on three orthogonal principles (i) derivation principle, which models the enterprise in three successive levels, (ii) instantiation principle which is also based on three levels from generic, via partial to particular and (iii) generation principle which suggests modelling the enterprise in four basic but complementary viewpoints.
- **Integrating Infrastructure** provides technology to achieve physical and application integration. Its purpose is to transform a highly distributed heterogeneous environment into an environment which looks centralized. Techniques traditionally used are OMG/CORBA for object oriented systems.
- **System Life Cycle**, is the way CIMOSA defines a generic life cycle. It constitutes a sequence of phases to be used as bricks to build a specific architecture. The major phases of the life cycle are (i) master plan definition, (ii) requirements definition (iii) system design, (iv) system build and release, (v) system operation, (vi) system maintenance and change and (vii) system dismantling.

For more and detailed information on CIMOSA and the framework which CIMOSA supports information can be found in [Kosanke et. al., 1999] [Kosanke and Zelm, 1999] [Ortiz et. al., 1999] [Berio and Vernadat, 1999] [Segarra, 1999] [Kosanke, 1995]

3.5.2 PERA

The Purdue Enterprise Reference Architecture (PERA) has been developed at the University of Purdue since the late 1980s. The methodology is based on work within the CIM area. PERA is characterised by the layering structure. It has been created to cover the full enterprise life cycle from inception and mission definition to operational level and final plant. Each layer defines a task phase. Each phase is described by an informal technical document which is a set of procedures for leading a group of users through all the phases of an enterprise program.

PERA does not provide a modelling tool of its own and therefore other existing tools can be used which provide the best modelling capabilities at the time. Even though PERA was originally developed for manufacturing systems, it can be used in all types of industries. For more and detailed information on PERA and the framework which PERA supports can be found in [Li and Williams 2000] [Ortiz et. al. 1999] [Williams 1994]

3.5.3 GIM

Originally GIM stood for GRAI-IDEF0-Merise, a methodology for analysis and conceptual design of manufacturing systems. However, the name has later been changed to GRAI Integrated Methodology. Both GRAI and GIM have been developed at the University of Bordeaux. GRAI and GIM are conceptual models and both of them are included in the GRAI concept. The concept GRAI points out is how any enterprise consists of four fundamental sub-systems, (i) a physical system which focuses on material flow, (ii) an operating system which focuses on real-time control, (iii) a decision system which focuses on the decisions in an enterprise and (iv) an information system which provides the link between the decision system and the physical system.

The GRAI method has been applied to a variety of manufacturing system design and methodology projects. The method uses two basic modelling tools which are the GRAI grid and the GRAI nets. More information on those can be found in [Doumeingts et. al. 1993]

The development of GIM has been strongly influenced by the development of CIMOSA from which it has borrowed some of its concepts. Further, GIM is based on three abstraction levels, namely: (i) a conceptual level, (ii) an organisational level, (iii) a physical level. They form a two dimensional matrix together with the three views of information, (Data, Process and Operational). This matrix has the same scope as the three dimensional matrix in CIMOSA and each cell can be populated by one or more models.

GIM also supports a structured methodology which aims at providing specifications for building a new manufacturing system. It is mainly focused on organisational aspects, information technology and manufacturing technology. The methodology includes four separate phases (i) Initialisation, (ii) Analysis, (iii) Design, (iv) Implementation. For more information on GIM and the framework which GIM supports, detailed information can be found in [Doumeingts et. al. 1993] [Vernadate 1996]

3.5.4 GERAM

GERAM stands for Generalised Enterprise Reference Architecture and Methodology. The purpose of GERAM is to serve as a reference for the whole community concerned with the area of enterprise integration. It should provide definitions of the terminology, a consistent modelling area, a detailed methodology and promote good engineering practise for the building of reusable, tested and standard models. It should also provide a unified perspective on products, processes, management, enterprise management and strategic management. In order to realise the above, GERAM consist of seven major components:

- a Generic Enterprise Reference Architecture (GERA)
- a Generic Enterprise Engineering Methodology (GEEM)
- Generic Enterprise Modelling Languages (GEMLs)
- Generic Enterprise Modelling Tools (GEMTs)
- Generic Enterprise Models (GEMs)
- Generic Enterprise Modules (GMs)
- Generic Enterprise Theories (GTs)

GERAM provides methodological guidelines for enterprise engineering (from PERA and GIM'), a system life cycle (from PERA), and constructs for modelling (e.g. CIMOSA constructs). GERAM is not yet another proposal for enterprise reference architecture, but is meant to organise existing enterprise integration knowledge [Bernus and Nemes 1997] [IFIP-IFAC 1999] [Vernadat 1996].

3.5.5 NIIIP

The National Industrial Information Infrastructure Protocols (NIIIP) Consortium is a group of organisations and companies with the interest of providing technologies to enable Virtual Enterprises. The goal of the NIIIP project is to solve incompatibility within Virtual Enterprises and allow organisations to collaborate with each other regardless of data structures, processes, or computing environments. The objectives of the NIIIP project are to develop, demonstrate, and transfer into use the technology to enable industrial Virtual Enterprises to collaborate and share engineering and manufacturing information.

Industrial Virtual Enterprises must be able to inter-operate and exchange information in real-time so that companies can work as a single integrated unit. The NIIIP consortium uses core technologies such as Internet and related communications facilities and services, Object Management Group (OMG) and related object technology, STEP and related information modelling technologies. Because these technologies are not well integrated, additional technology is needed for work and knowledge management of Virtual Enterprises [NIIIP 1998].

The NIIIP Consortium developed a reference architecture and prototype implementation for the technology needed for industrial Virtual Enterprises. It consolidated, harmonised, integrated, and extended existing protocols. There are four technology requirements for industrial Virtual Enterprises [NIIIP 1998]: (i) Common communication protocols, (ii) Uniform object technology for system and application interoperability, (iii) Common information model specification and exchange and (iv) Cooperative management of integrated Virtual Enterprise processes.

The design goal of the NIIIP reference architecture is a software architecture that enables Virtual Enterprises to synthesise any collection of resources and technologies into a production system, similar to a stereo system, where users can select and connect different components and make them work as an integrated unit.

3.6 Summary on Information Management and Standards

The present chapter has surveyed a vast amount of materials considering information, data and standards. With the aim of explaining the difference between data, information and knowledge this chapter also reviews standards that are used in information management.

This chapter discusses the importance of rules in communication between different software's. The information transference builds on a receiver and a transmitter of data which is sent according to a predefined set of rules. If the rules are omitted or not identical at both parties data will be lost.

Data management systems utilise the internet and related technologies to distribute data over a larger area. Several systems e.g. PDM, MDM and ERP, are based on the same technology but they differ in the information they contain.

ISO 10303, STEP, is one of the important standards that cover a large part of the engineering domain. Even though the initiative to the standard came over 20 years ago, there is still much to be done. By combining ISO 10303 with other standards and initiatives there are possibilities of supporting data management systems with standardised information models.

4 Requirements on a Framework for Effective Information Support of Manufacturing Simulation

During the development of DES and other tools used for conducting simulations, focus from the start has been on shortening the modelling phase. Traditionally the modelling phase (data collection and model design) was about 60% of the total time spent on a simulation project [Umeda, 1997]. In the mid 1980's simulation was considered very expensive and only used in very special cases where the benefits far exceeded the costs associated with its use. Furthermore, computational power was at the time fairly limited compared to today and simulations were thus very rarely run on a desktop computer. Instead, simulation systems were run on mainframes, which both are very expensive and demand much monitoring. This delayed progress in the development of simulation tools for the market. However, there were enthusiasts who were using and developing simulation systems and modelling methodology, in order to improve the tools in which they saw great potential.

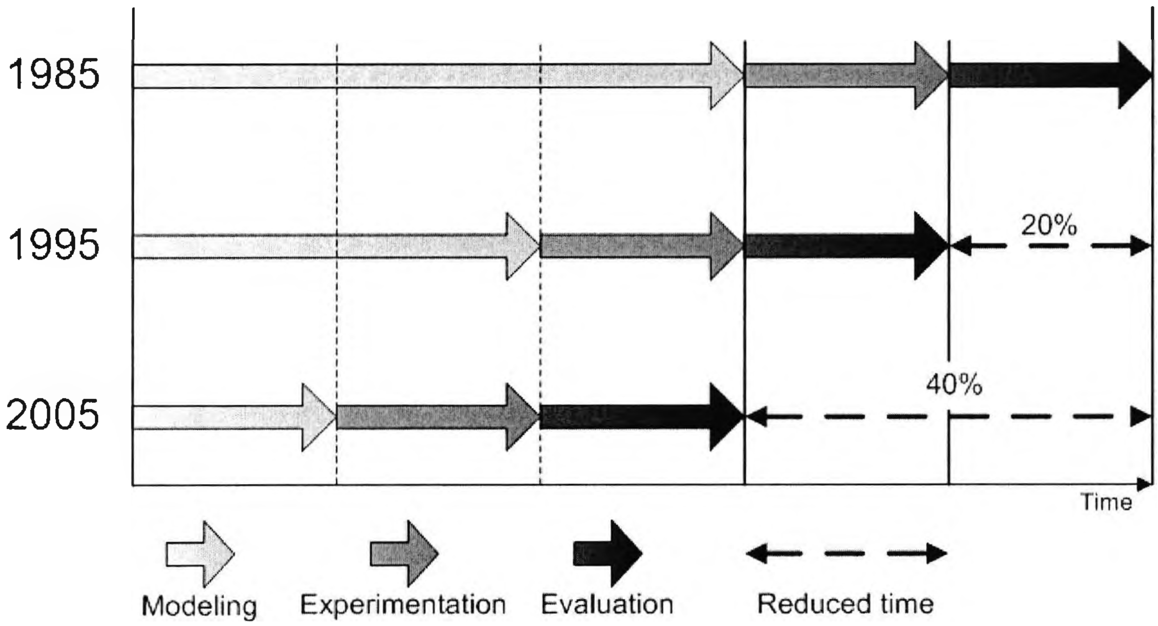

Figure 14 Reduction of modlling time, modified from [Umeda, 1997]

In the mid 1990's the modelling phase was considerably reduced, see Figure 14. This was mainly achieved by a strong focus in this area the past decade. Further, much more of the data needed to conduct simulations was stored in digital format, e.g. CAD data. The growth of CAx products on the market has played an important part in the development during this decade. Many of the CAD-system manufacturers were supplying CAx integration possibilities in the mid 90's. The development of PDM and other data management systems made quite an impact on the product development. Many in the simulation community were thus provided with a vision of integrating the information sources with the simulation systems in an effort to further reduce the modelling phase.

Today there are several attempts on the market which claim that they have solved all the problems of integrating the information sources with the simulation systems. The modelling phase is now shorter than ever, providing that the information integration is working, see Figure 14. However, the attempts on the market are not based on international standards and they do not easily allows integration of simulation software and information sources from different manufacturers. That has made it harder to maintain the short modelling phase which is needed to make DES a tool in the toolbox of an engineer.

A number of requirements can clearly be identified in order to fulfil a generalised framework, which is successful in supplying data to a simulation system. The framework also has to make the modelling phase as short as possible, in order to ensure that simulation is as profitable as possible.

4.1 Simulation System Requirements

There are many different simulation tools on the market that proclaim to be the best at solving all the problems. However, the truth is that different systems are the perfect one for different organizations. In order for a simulation tool to be considered a tool capable of supporting effective information management some requirements have to be met.

4.1.1 Easy to Model a System

In the modelling phase an average of 60% of the simulation project time is spent building the model including the data collection. This of course does not encourage the use of simulation for many projects because the modelling phase too long. It means that simulation is a tool both time and resource consuming and therefore only used occasionally for specific purposes. However, if the models in the simulations are built on standard components, the time needed for building them could also be reduced. Some simulation packages have developed special libraries for certain types of industry for this reason.

4.1.2 Easy to Use

The advances in technologies during the last decade have enabled simulation to become much more user-friendly. It is now possible to run a simulation on your desktop or even laptop computer with a 3D animation of your model. Simulation is more than a tool for specialists, and to bring simulation to every desk in the company (managers, project leaders, and engineers), a short training program and a handbook for conducting simulation projects have to be provided [Jägstam, 2004].

4.1.3 Easy to Re-Use or Update Models

One of the most frequent limitations mentioned by companies using or planning to start using simulation tools is that it takes too much time to update simulation models when a change occurs. There are seldom any larger libraries of models supplied with the simulation packages which could be used to rapidly build new or alter existing models. The need of re-useable models is driven by the fact that companies are constantly required to adapt their production to market demands, which leads to a constantly changing production environment. Different solutions of the production facility need to be evaluated, and one way of solving the problem fast is to have a library which contains models of each component of the enterprise system. Another alternative is to make the components easily reconfigurable with data from the production system linked to the components. Both alternatives can greatly reduce the time spent developing a new model.

The latter alternative is further explored and discussed in the following sections.

4.1.4 Easy to Integrate with Other Systems

As stated above, simulation is today made available on a normally equipped desktop computer, assisting part designers, manufacturing engineers and manufacturing managers in individual decision-making. However, to support a concurrent development process and Virtual Manufacturing, simulation should become an integrated activity of the product development phases [Bolmsjö et. al., 1999] [Ng, 2003]. To provide the required communication for that objective, simulation tools should be integrated with the different systems of the enterprise. Depending on the level of the simulation study, different systems are to be used. Some of the systems are listed below:

- PDM system
- ERP system
- CAD
- CAPP system
- CAM system
- PLC system
- Monitoring systems
- Other simulations packages

The integration of simulation tools with information sources makes the gathering of data for the building of the simulation model much faster but not always easier because of the large amount of data. There have been arguments in favour of automating the data collection when the information sources can be integrated with the simulation system. But making the data available to the systems on a technical level is only one part of the integration. Selecting the data to be used is still difficult to automate, even if there have been attempts to develop smart programs which to some extent can judge if the data is interesting or not. However, a more difficult scenario than deciding which data to use is when the same data is present in several different systems at the same time. It is not uncommon that there are slight differences in the data and that the original source can be hard to find. Then a person has to do the selection based on knowledge and strategy.

4.1.5 Selectable Degree of Abstraction of the Model

One of the advantages of using simulation when collaborating with suppliers within the product and production facility development process is the possibility of visualising the area of interest. In order to use the advantages with visualisation, the modelling has to be done as the right level of abstraction.

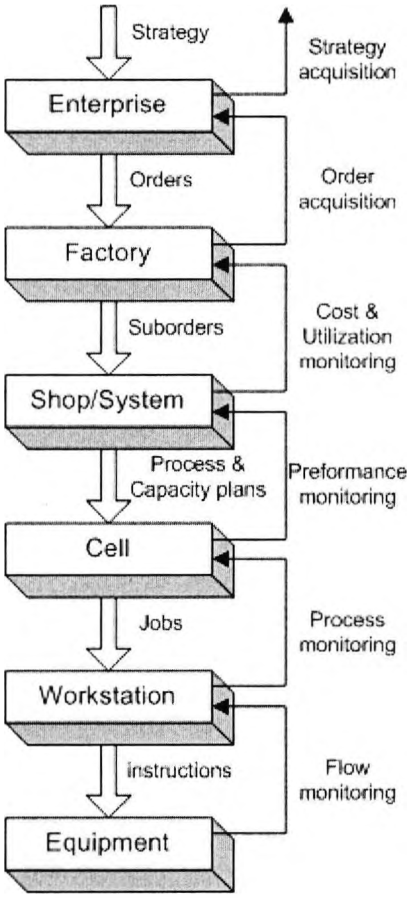

Figure 15 ISO decomposition of an enterprise [Bauer et. al., 1991]

The ISO model of an enterprise decomposition, see Figure 15, illustrates that an enterprise can be described at six different levels. Using this decomposition model together with simulation, the level of abstraction can clearly be identified. In practise it means that results achieved in simulations at a lower level are integrated in the higher levels.

Furthermore, different modelling techniques have to be adopted at the different levels of abstraction. At the enterprise level mainly low detailed models are used. The graphical interface is less important, since any further understanding can seldom be given the user through that media, However the data output is more important and often has to be plotted in diagrams for the management.

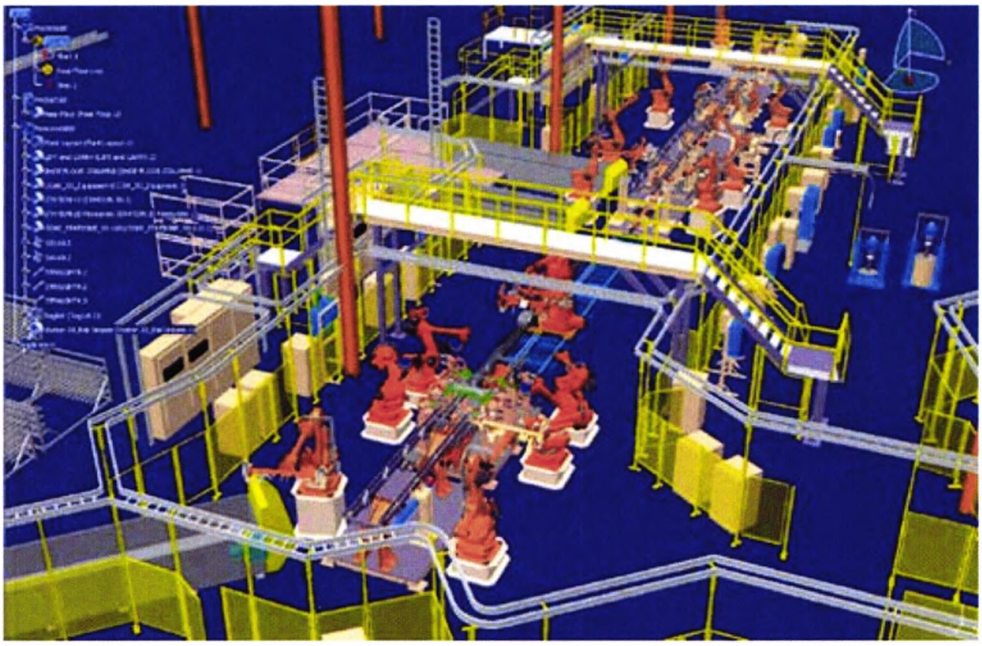

Figure 16 Factory level of simulation model [Delmia, 2005]

The factory level, Figure 16, is generally described by a discrete event simulation model which logically connects all the resources in the factory in order to study dynamic effects when they are put together in a system. The line level, Figure 17, describes a selected area in much greater detail. Mostly DES tools are used, but also other types of simulations can be seen as alternatives depending on the type of industrial process.

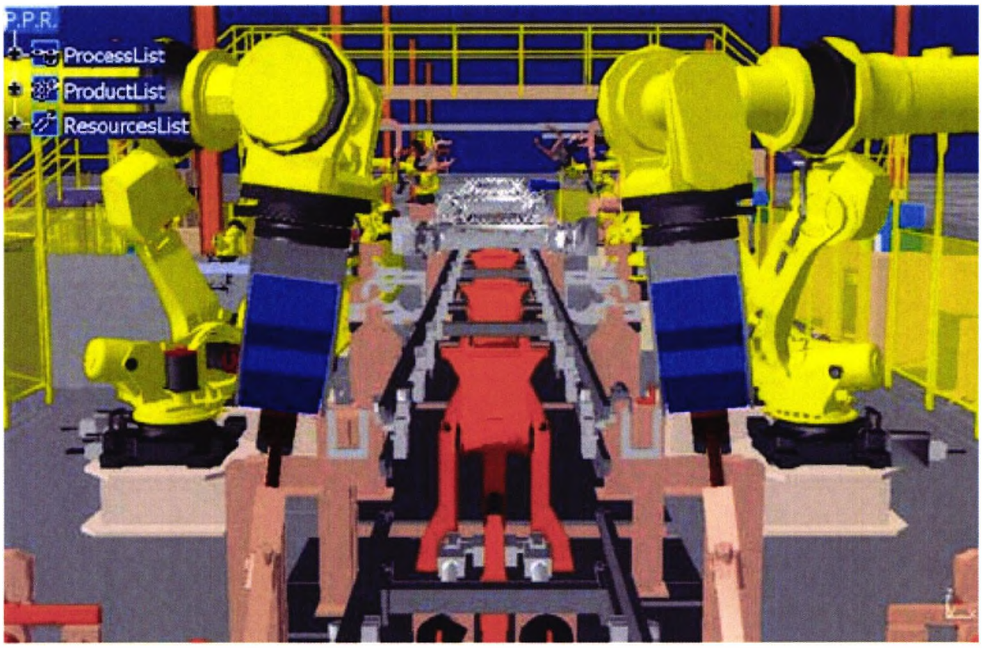

Figure 17 Line level of simulation model [Delmia, 2005]

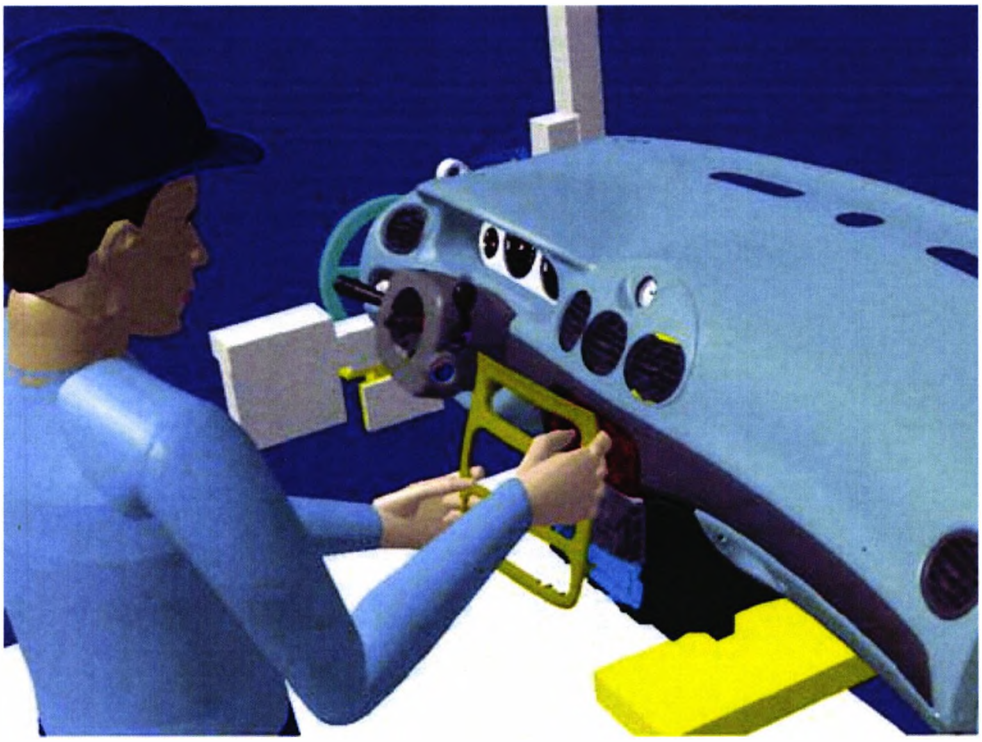

Figure 18 Cell level of simulation model [Delmia, 2005]

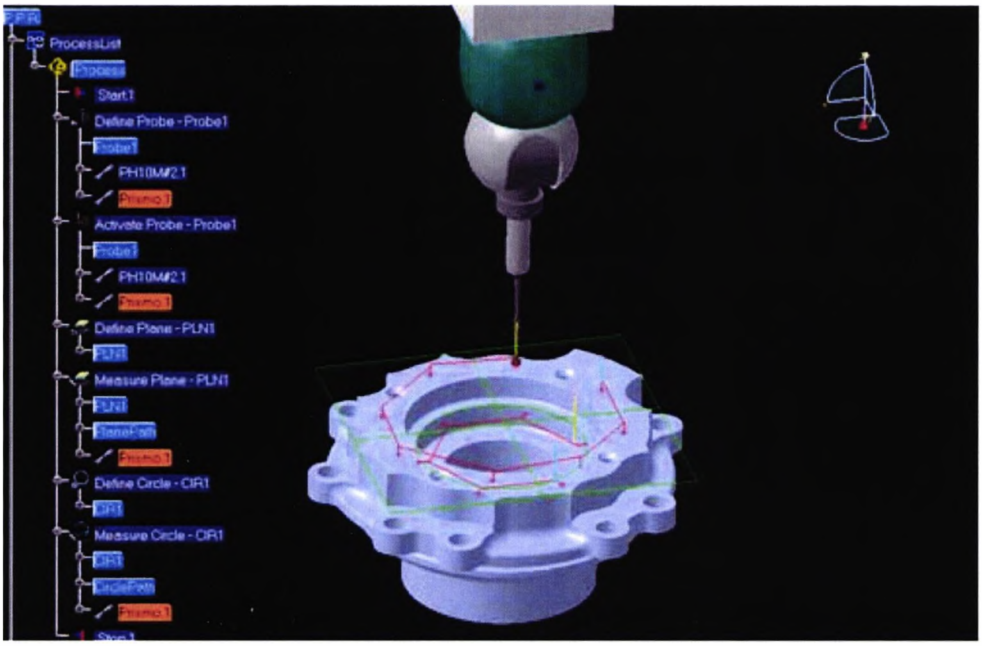

Figure 19 Unit level of simulation model [Delmia, 2005]

A more detailed study of some of the cells that are included in a line may be needed, see Figure 18. Such a study often involves several different types of simulations. The simulations required are typically from several different areas necessitating a lot of special knowledge in each respective area. A typical study is ergonomics simulations, where an average workplace could be analysed to understand the everyday motion patterns and how they affect the body.

The lowest level of simulation is at the unit level. Sensor simulation [Eriksson, 1996] is one that can be conducted on this level. It typically illustrates how one unit behaves in correlation to another. Simulations of fixtures and jigs are other examples conducted at this level, see Figure 19.

4.1.6 Accuracy of the model

Not only the tool has to be reliable. The data put into the system is what the model has to work with. A thorough and correctly performed input analysis must be conducted on the data available. The accuracy of the model is also related to the level of detail. A more detailed model has the chance of being more accurate although one should not just assume that the greater the detail the more accurate the model.

4.1.7 Affordable Price

Unfortunately, cost has been one of the major obstacles to the development of simulation in industry. To implement simulation, several cost factors have to be considered [McLean 1999]:

- Computing hardware and peripheral devices
- Software licenses and maintenance
- Simulation and support staff training and salaries
- Search for and input of relevant data
- Translation of existing company data
- Information integration of applications
- Development and maintenance of models

Even though the price of a simulation software package (a license) is the first cost a customer evaluates, it is merely one part of the simulation budget. Most of the other costs, such as finding data that is relevant and accurate, the translation of company data, or information integration of applications are directly linked to the capabilities of the simulation software purchased. The importance of proper data management and the integration of the simulation system with information sources are the most expensive parts following the initial cost of buying the simulation package. These costs cannot be eliminated but can be reduced considerably by planning ahead and selecting a system which contains the right data management and integration functionality. [McLean 1999]

4.2 Data Requirements

Four different requirements for information systems in extended enterprises, to which digital manufacturing solutions belong, have been identified. [Al-Timimi and MacKrell, 1996] described the following four requirements: interoperability, portability, longevity and extensibility. Furthermore, scalability is an additional requirement for information systems in enterprises that are undertaking virtual manufacturing.

4.2.1 Interoperability

Different applications must be able to share information in order to ensure that all users work on the latest information available. To enable this information sharing all involved systems must have a common definition of the semantics of the information they manage.

Further, team level interoperability can only be achieved by utilising common definitions of the work processes involved. It means that a common definition of workflows and the flow of information between processes has to be developed and used. [Kemmerer, 1999]

When implementing common definitions of information and workflows the possibility of implementing information management mechanisms that potentially can support the execution of workflows is provided.

4.2.2 Data Portability

Typically, data that describes products and manufacturing systems must be able to be shared and exchanged within the domain of application that constitutes the platform of digital manufacturing. The information to be shared must be in an application neutral format which can be utilised by all applications. Portability is one of the means of fulfilling interoperability.

4.2.3 Longevity

The information an application has generated should outlive the computer platform on which it was created. That is important in order to fulfil the increased requirements on re-using information from pre-existing designs. The main advantage is the possibility of increasing the speed of developing new products and manufacturing systems. Moreover, the information that different systems generate needs to be maintained, so that it can be utilised after the systems that created it are no longer available. To achieve longevity data semantics must be application and version independent.

4.2.4 Extensibility

Information in the digital manufacturing platform needs to cover all thinkable properties of the described objects in order to provide extended application functionality. This may be very difficult, or in fact impossible to achieve. An acceptable solution is to ensure that information is developed in a way that permits the possibility of easily extending the information to fulfil additional information requirements.

Extensibility is of major importance in order to produce solutions for information representation that allow changes in design and modelling techniques. It provides the possibility of using new development tools and technologies.

4.2.5 Scalability

The implemented digital manufacturing solution must be prepared for a continuously increasing number of users. Therefore scalability is of interest for constructing the information systems that have to handle the growing amount of information and the increased complexity which will be the consequence thereof.

4.3 Architecture

With DM-solutions a solution has evolved which has proved to be very useful in large systems where the users of the systems are located far apart. The solution is called "*three-tier architecture*", after the separation of the information, Figure 20.

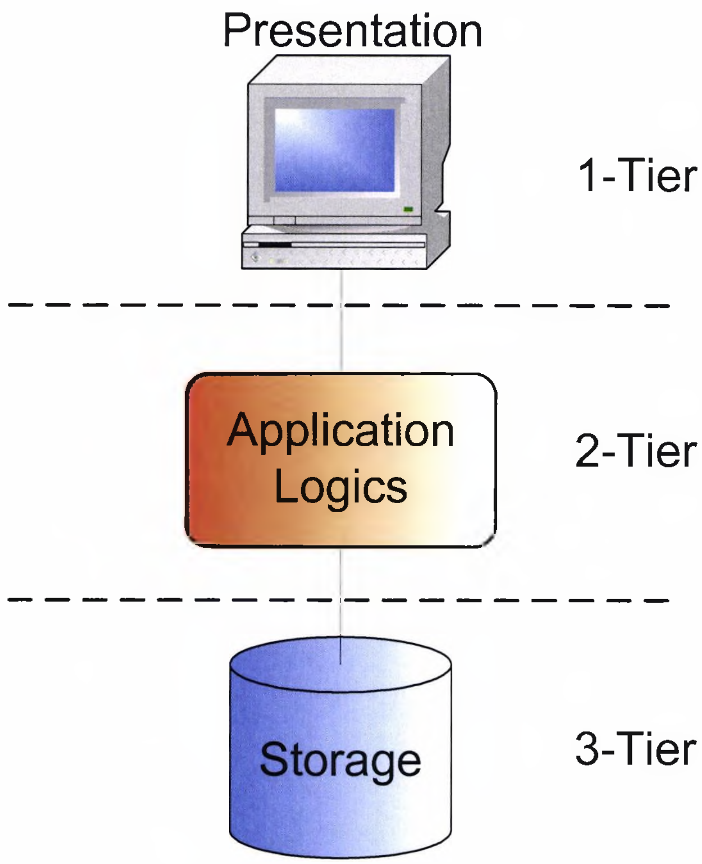

Figure 20 Three-tier architecture

The separation of the tiers enables information model transparency by decoupling the information model in the application logic tier from the application processing. The application tier communicates with the back-end storage layer.

The three-tier architecture enables scalability of the information systems. This is managed by offering the possibility of continuously broadening each tier in the system independently. Further, not all tiers have to be physically placed on the same computer. For instance, the application tier can with advantage be physically located elsewhere in the network. To handle the connections between the tiers CORBA or DCOM are utilised. Those techniques have many advantages in environments where distributed objects have to be managed in an efficient way [Olofsgård, 2005].

The three-tier architecture is today implemented in all PDM systems on the market [Liu and Xu, 2001]. It has been shown over time that the architecture has the capability of managing the growing number of dislocated users in ways no other techniques are yet capable of. However, the architecture can be illustrated within PDM-systems with the following. CAD-systems, DES-systems, etc. and other types of applications are in the application tier. The metadata that contains references to the applications and the data stored in the system are located in the application logic tier. This is also commonly named the method server of the system. The network file servers hosting the referenced application files operate in the storage tier.

Aganovic [Aganovic, 2004] summarises three different types of industrially well known solutions for the three-tier systems that have different capabilities in performing data integration with other systems which is not included as an application.

4.3.1 Type 1

The first solution provides a common information platform that is equipped with external references to share information between their own applications, see Figure 21. Further, the lack of workflow capability in combination with a proprietary information model limits the interoperability to a minimum. This includes that there is no workflow control present, as well as no or very limited interoperability for applications outside the system.

The propriety information model in combination with the lack of standardised interfaces such as STEP precludes data portability. The lack of interfaces also limits the ability of commenting on the longevity of the data and the extensibility of the information model. Assumptions that can be drawn from this, is that both longevity and extensibility is limited due to the lack of transparency in the solution domain.

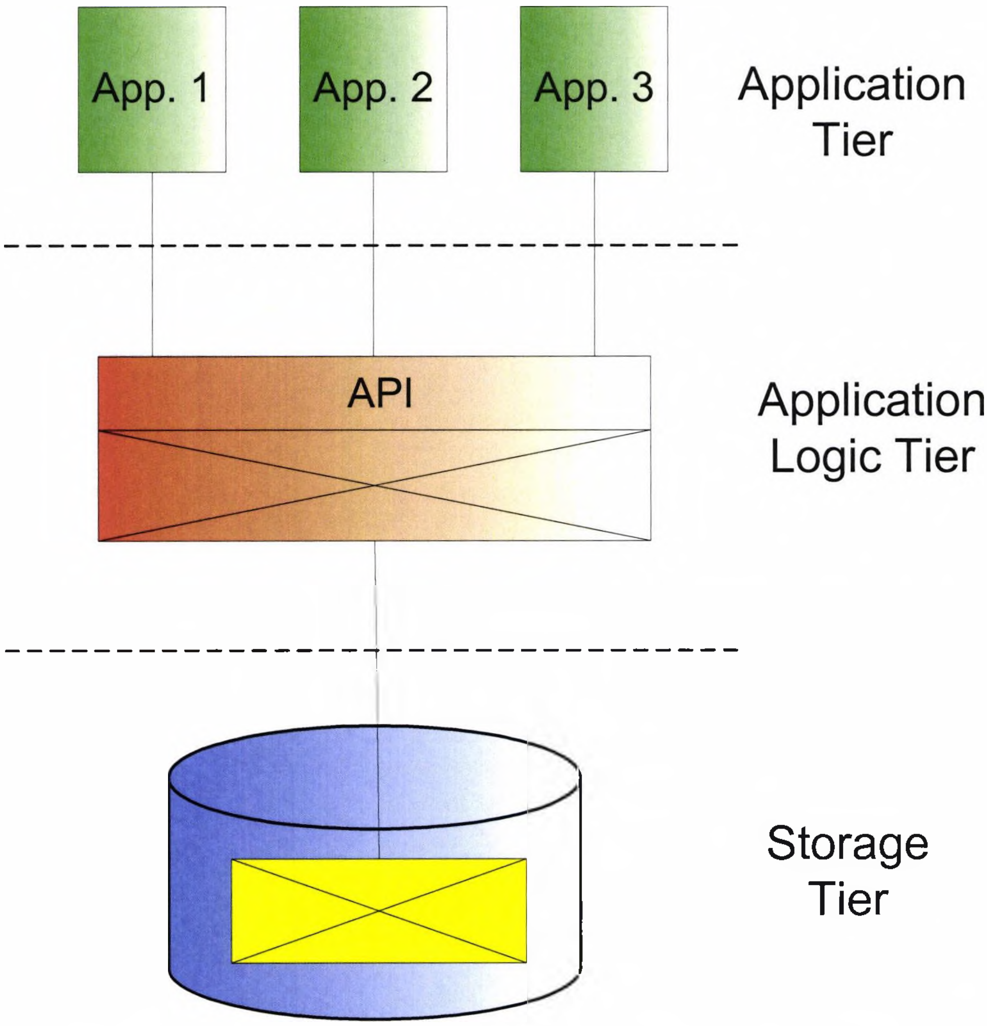

Figure 21 Simplified architecture of solution 1, adopted from [Aganovic, 2004]

4.3.2 Type 2

This solution provides a common information platform, with external referencing and workflow capability of sharing information between the applications that are included in the system, see Figure 22. Compared with solution type 1 the information model of solution type 2 is proprietary. This provides interoperability for the system applications, such as workflow control and sharing of data, but it also imposes limited interoperability for applications that are not included in the system, 3rd part applications.

Solution type 2 has similar to solution type 1 no data portability and no or very limited data longevity and information model extensibility.

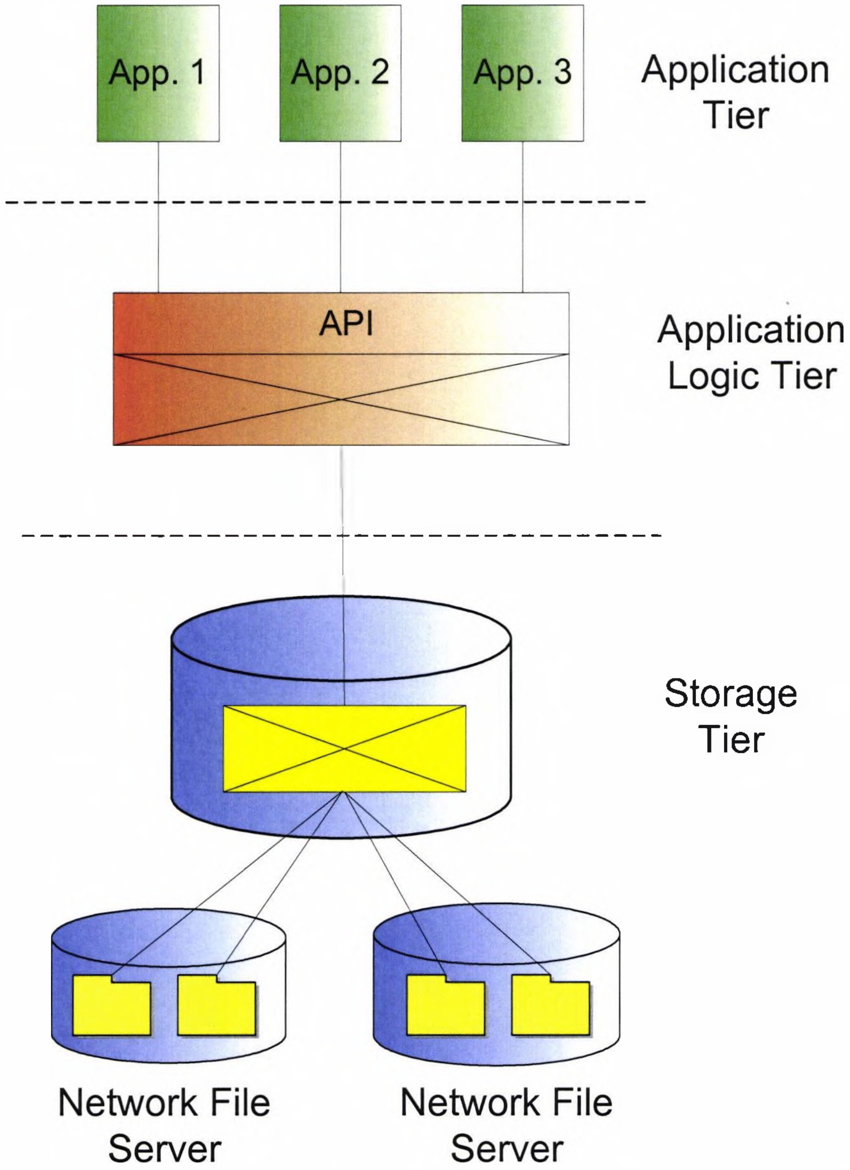

Figure 22 Simplified architecture of solution type 2, adopted from [Aganovic, 2004]

4.3.3 Type 3

Solution type 3 provides a common information platform with external referencing and workflow capability, to share information between their own and the 3rd part applications, see Figure 23.

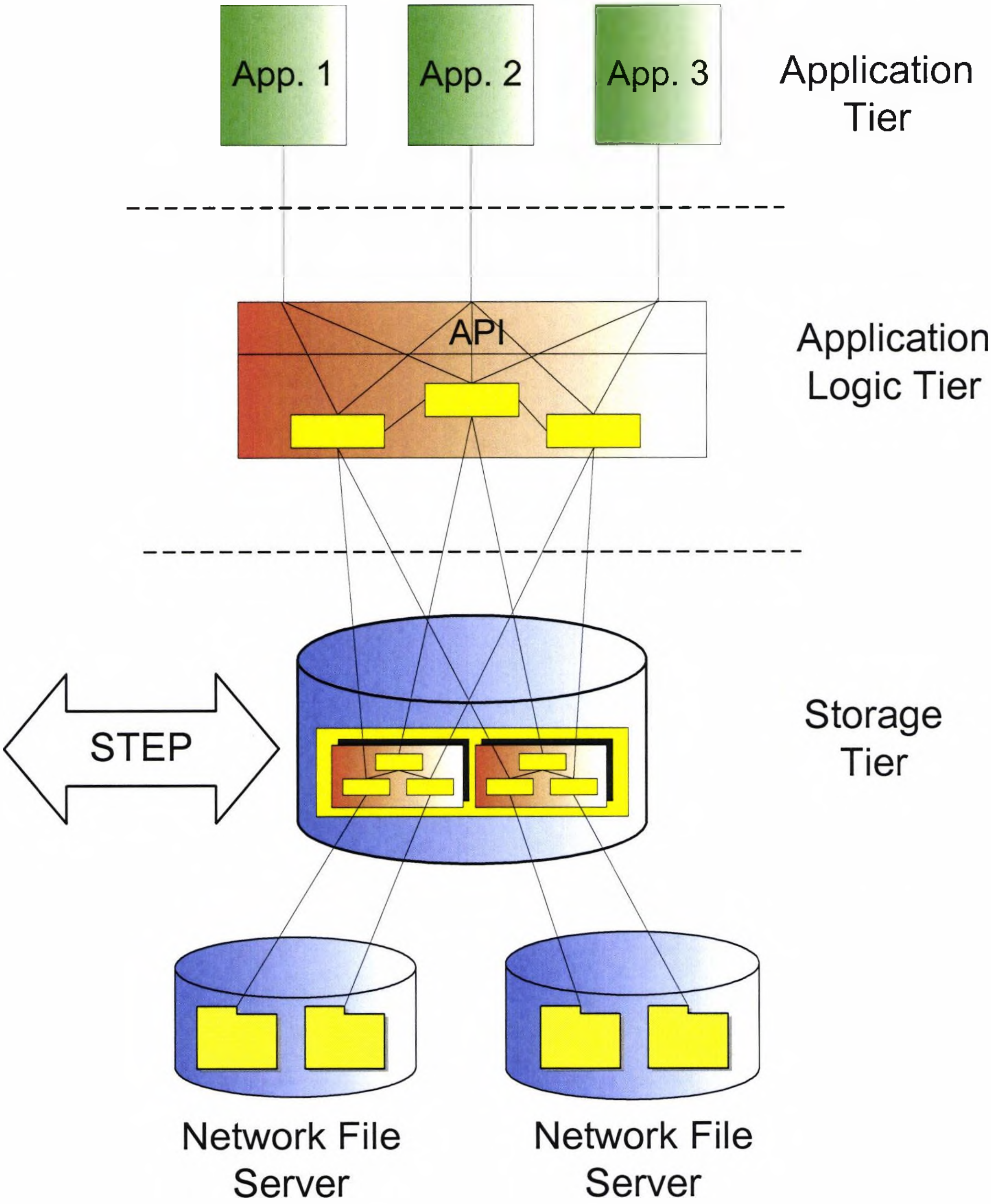

Figure 23 Simplified architecture of solution type 3, adopted from [Aganovic, 2004]

In contrast to the two former solutions, solution type 3 has an open or partly open information model. The more open information model in combination with workflow capability provides interoperability for the 3rd part applications that are incorporated in the system.

In addition, the open information model complemented with the support of a standard information data interface, such as STEP, as a part of the open information model, provides good means for data portability. Further, an open information model does not automatically mean data longevity and information model extensibility. The information model provides transparency to the solution which will make migration of data easier if a change in the information model should occur.

When an interface is based on international standards, such as STEP, it also provides stability and longevity to the solution. Even though the internal model changes, the standardized interface will remain the same as long as no major change is made to the standard.

4.4 Suggested Concept for Simulation Model Reuse Based on the Identified Requirements

The concept that is shown in Figure 24 proposes the use of the DM-technology to support the simulation activity. The objectives of the approach is not to modify the traditional phases in a simulation project but to merely provide a better support during the time consuming parts at the beginning of a project. The concept consists of three phases which are fairly distinct:

- Data Management
- Modelling
- Validation and analysis

The approach in this concept suggests that most of the modelling of the simulation system is done outside any simulation tool. The modelling activities are supported by the distributed data sources found in traditional DM-systems, but also from other systems that handle data which manages the manufacturing systems.

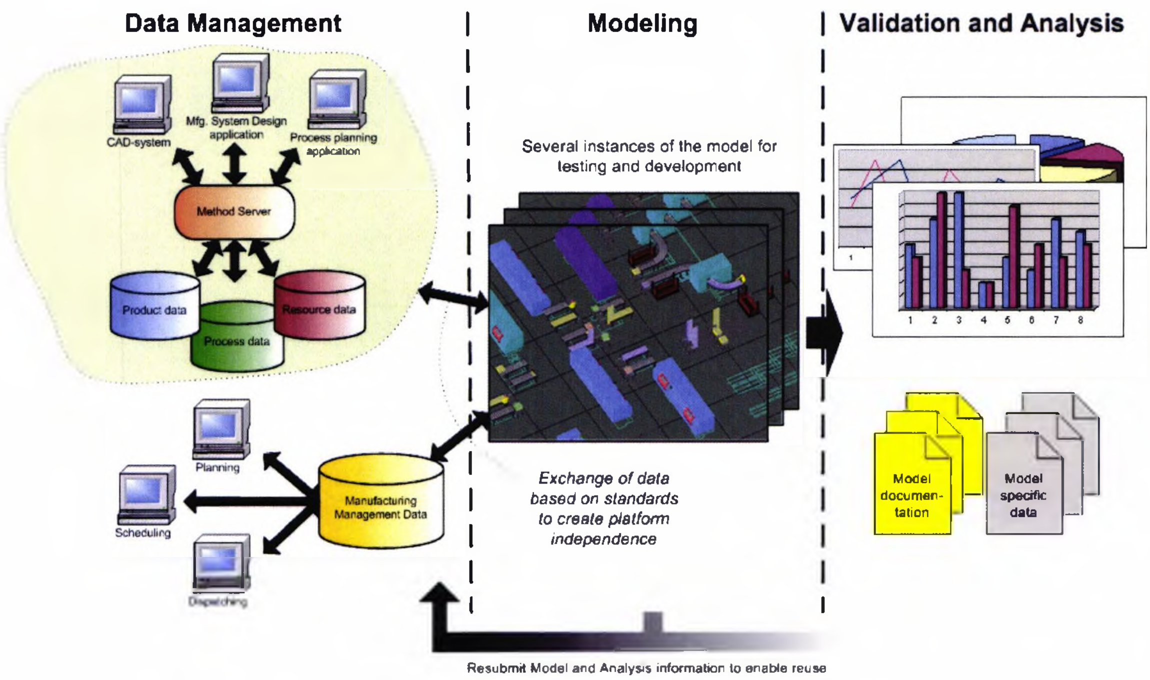

Figure 24 An information support concept for simulation

Ultimately, with this approach and concept the simulation technician would not have to collect the data from all kinds of different sources but instead only select the information that is needed for the study. However, that can only be feasible in an ideal world. Instead, the concept provides a support and opens a possibility of extending the data support for simulation studies and to minimise the data sources that have to be taken into consideration.

Further, during the modelling, validation and analysis phase a simulation tool of choice, which is also appropriate to the objective of the project, i.e. ergonomic simulation, process simulation, or manufacturing flow simulation, can be utilised by the simulation technician in order to perform the execution of the simulation model. The support from the information sources is based on standardised interfaces which provide the information in formats that are supported by international standards. This constitutes a good solid base for a reliable supply of data over a longer period time.

Figure 24 visualises the concept and illustrates how simulation tools can be integrated with information sources, which allow engineers to spend less time on data collection and modelling. However, it increases the amount of time spent on the evaluation of different scenarios of the system, which can be considered as value-added work in a simulation project.

4.5 Summary on Framework Requirements

Reuse of data and models for virtual manufacturing simulation demands a number of different requirements, which are identified from both the domain of simulation system, data format and availability. It can be concluded that both data and simulation system requirements have to be fulfilled in order to achieve a good system.

Three different types of information systems architectures are described. They are based on a three-tier structure but with different mapping of the data towards the storage tier. They also differ in their ability to handle external data format such as STEP.

At the conclusion of this chapter a concept based on the requirements and the architectures is presented. The concept utilises several information sources to support the modelling and analysis as well as providing a feed back loop to the information systems with the result from a selected scenario.

5 Concepts for Use and Reuse of Information for Manufacturing Simulations

During an industrial analysis of the data collection process, Robertson and Perera, [Robertson and Perera 2002] identified four different methods to input the required data in to the model. These are detailed below and also illustrated in Figure 25. Robertson and Perera, 2002 came to the conclusion that the methodologies "a" and "b" in Figure 25 are currently used extensively within the industry, whereas the methodologies "c" and "d" are only just emerging. As such, there is only one isolated case evident for each methodology within the industry. Robertson and Perera predict that the methodologies "c" and "d" will offer great advancements with respect to data collection in the future, although it is dependent on their intended application type [Robertson and Perera, 2002]. Details of these approaches are provided below:

a) The model builder manually collects the required data via various mechanisms such as data templates completed by the project team, using information spreadsheets or interviewing individual domain experts. The data is then manually entered into the model as and when required. This is a simplistic method, especially in larger manufacturing organisations that should be leading the way. The benefits are that it is a simple method to follow for model building, and the modeller verifies all the data as it is entered into the coding. However, it has inherent drawbacks, principally due to its manual nature. These are namely the extended time, effort and errors. In addition, the data is stored within the modelling tool and it is therefore a very inflexible system, as the coding will need to be changed if any of the data is modified.

b) As in the previous methodology, similar mechanisms are used to collect data, but this data is then combined from the various sources onto a formatted "master" spreadsheet. This spreadsheet contains the majority of the required data for the model. Once the model is built, the data is automatically imported from the spreadsheet to the model. Hence, the data is stored externally in the model, which enables flexibility of the data and the model. The in/exporting of the data required by the model is a significant advantage on the previous method, but it is time and effort consuming to combine and format all the data. This methodology is becoming an increasingly popular method within the industry.

c) The model utilises an Intermediary Simulation Database that automatically retrieves and stores the required data from the sources within the Corporate Business Systems. The model "reads" the required data from the "integrated" database to run the model. Again, the data is stored externally to the model, thereby introducing flexibility. As the data becomes available from the Corporate Business System, the intermediary database gathers and manages the model data. Thus time, effort and errors can be radically reduced if the intermediary database automatically refers to the data in the Corporate Business System. This method is not apparent within the industry. A major US car manufacturer has adopted this methodology. Although the intermediary database shown in method "c" is in fact a spreadsheet that is populated by the corporate business system.

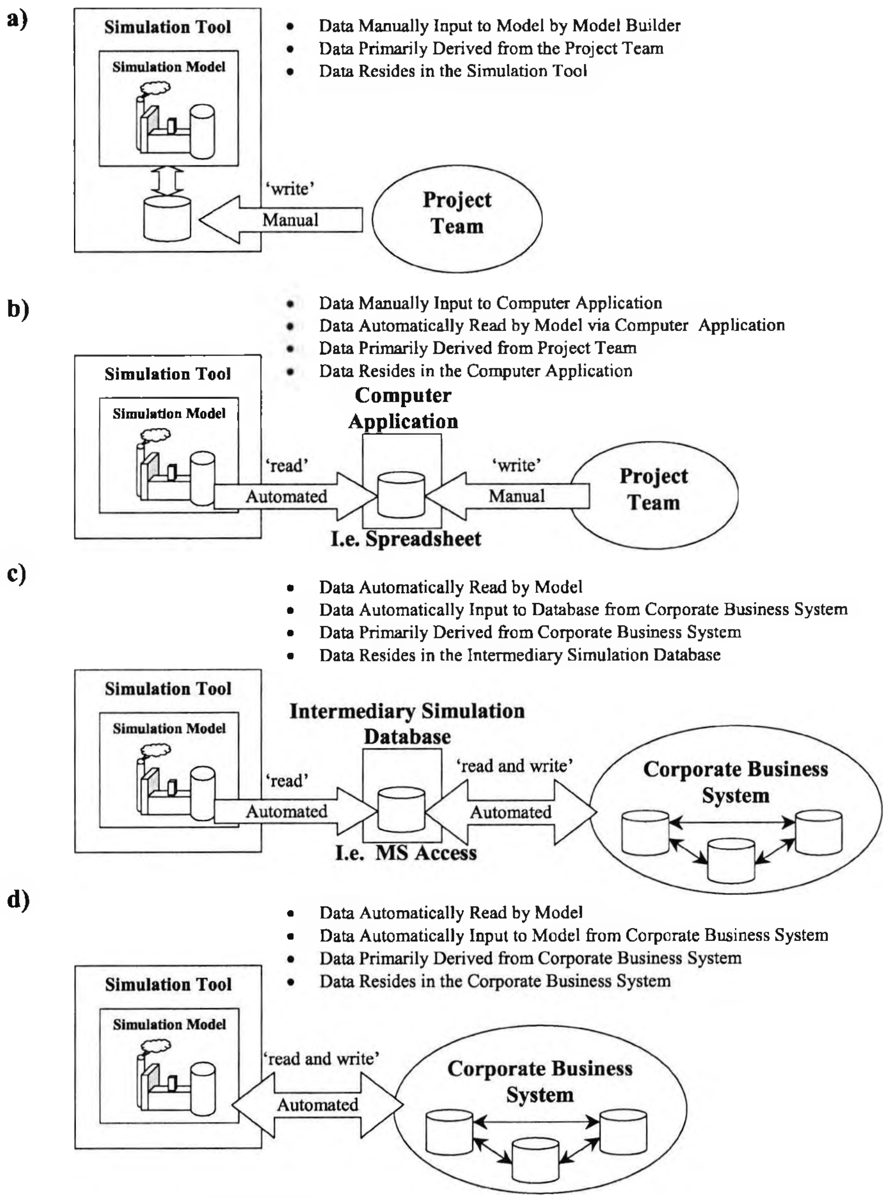

Figure 25 Possible data collection methods for model building [Robertson and Perera, 2002]

d) In this methodology, the model automatically collects data from the Corporate Business Systems via an interface as and when required in order to run the model. Essentially the model is referred to an external location to "read" the data directly from the Corporate Data Systems. Again, the data is stored externally to the model, thereby inducing flexibility. In addition, as the model is built, it is referred to sources within the Corporate Business System. The automated referral system dramatically reduces time, effort and errors. There is a major drawback due to the complexity and size of this methodology. As there are so many different sources there will inevitably be alternative sources for the same data item. This data duplication may cause a problem in terms of data accuracy, reliability and validity. A further hindrance is that in some situations the data required by the model may not be available at the source when the model refers to it. This methodology is extremely complex to establish and it is apparent only in one isolated case within the industry where it is used for System Management as opposed to System Design. This methodology has been accomplished by a major US aerospace company, which has developed an interface that relates directly to the business system and the model. It is used for System Management since it is used for routine modelling of the production line on a daily basis. The methodology rapidly modifies the production line when circumstances change, such as amendments to the schedule provided by the corporate business system. It then recommends actions to be taken in order to meet the amended requirements. This set-up is more suited to System Management rather than System Design, as the possible configurations of the system are already known, simplifying the modelling process for System Management.

It is apparent that the methodologies ("c" and "d") in Figure 25 which integrate the simulation model to the Corporate Business Systems could be the solution for the data collection process, as it would increase the data accuracy and reliability while also minimising data collection efforts. This integration is extremely difficult to establish, but the long-term benefits such as easier data collection, increased data accuracy, reliability and validity, decreased data duplication, and speedy data availability are all profitable advantages.

The Corporate Business Systems mentioned have in recent years evolved to be incorporated in to a single extended information management system. These systems are particularly advanced in larger industries, where such systems are very complex but highly beneficial.

Due to their complexity and the level of detail required for model building, it is acknowledged that not all of the required information will be available through one medium. Hence, several software tools are commonly used, in order to complete the corporate business system. [Holst , 2001] [Robertson and Perera, 2002]

5.1 The Building Blocks of Information

In order to handle all the data that VMS applications requirements there has to be a structure, which illustrates how one bit of data relates to another bit of data. The data must have a format that is interpretable by using a stringent semantic, or it will be almost impossible to have any type of computer interpretation of the data. Last but not lest, data for VMS must be put into context, given a meaning, or there is no advancement. All these requirements of the data for VMS, transform the data into information, which namely comprises three ingredients; context, semantics and structure.

5.1.1 Context

Context is a very difficult topic since it depends on the way it is used in language. In order to understand context, it must first be defined. However, defining context is difficult since it is used instinctively. Most research on context can be found in the field of linguistics. However, some research has been done to breech the gap between the field of linguistics and engineering science. [Mills et. al., 2001] describes the role of context and several different types of contexts.

Context in VMS is decided more by the application in which the data is processed. The back end of the application that interacts with information supporting systems often has to work through different middle ware. The difficulties of transferring data in several stages are that a context which was very clear at one stage is not as clear in another application due to the internal structure and methods of work.

5.1.2 Semantics

Engineering tools, which are gradually becoming more computerised, have to deal with semantics at a very advanced level. It is not the engineer who controls the meaning of the information that is put into the system. Information semantics of computerised tools are based more on the semantics of the application designer, who has not always followed a formal format. In the past this problem has been in focus in the CAD environment. Different standards have been developed in order to transform something from one system to another. This emphasis has been beneficial but not without problems. [Kemmerer, 1999] [CIMdata, 2001]

However, STEP has proved to be beneficial in more than the area of CAD. STEP is a very good support for manufacturing and information sharing in general thanks to the rigorous definitions that the standard includes and the development of different areas which the standard covers. STEP's importance as a semantic tool is increasing in tact corresponding to the parts of STEP achieving the status of international standards.

5.1.3 Structure

Several systems for managing data have been developed over the past years. In those systems there have been some mechanisms that handled the structure of the information. There are several different database technologies in the area of database technology that have been developed for improving the performance of data retrieval and data structure. Object oriented and relational databases are the most common ones, but the technologies behind the techniques have been somewhat more difficult to use on their own. With the growth of the web and the use of HTML the ability to structure data for presentation became very easy but still not stringent enough for manufacturing application. HTML which has its origin in ISO 8879 SGML, had enormous potential to develop into something useful for structuring data. XML has evolved into a superior method for capturing data in a tree structure. That is because XML implements the foundations of SGML better than HTML. There is also the possibility of applying rules to a XML document, which makes it interesting in many ways. [Tian el. al., 2002] [Szykman et. al., 1999]

5.2 Three Domains of Information

In general, all information which is used in manufacturing simulations can be categorised into information domains. The traditional way of handling information has been to categorise it into domains which are clearly identifiable within an enterprise, see Figure 26. Three identified domains contain information about the product, the manufacturing process and the manufacturing resources. This separation of information into domains is also shared by [Azari,1990] [Eversheim et. al. 1991] [Eversheim and Westekemper, 2001] [Johansson, 2001] [Kulvatunyou and Wysk, 2000] [Onosato and Iwata, 1993] [Young et. al. 2000]. These domains can also be identified within the structure of ISO 10303-214 [ISO/TC184/SC4, 2001] and other standards within the area.

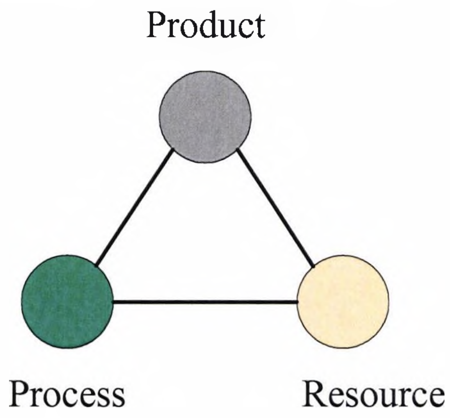

Figure 26 Interrelations between Product, Process and Resource

While each domain is required to capture its own design information and rationale in its own information model, as the different domains interact, it is important that design information about, and design rationale behind the interaction can be captured as well [Krause et. al., 1993].

The interaction between the different information domains could provide a description of the products, how they should be manufactured and what manufacturing resources should be used. That would provide an information platform upon which several different computer based tools to support the innovation process can be built. However, before understanding how the domains should interrelate with each other, an explanation of each information domain is required.

5.2.1 Product Information Domain

A product model represents all the relevant information about a product throughout its life cycle [Krause et. al. 1993] [Peltonen, 2000]. However, there are some differences in the definition of what should be included in the product information. On the one hand there is the definition only including the information which directly describes the product itself. On the other hand there is a definition which includes information describing how and with what resources a product should be manufactured [Krause et. al. 1993].

Naturally, some different aspects of several different life cycle stages must be represented in a product model. In general, there are aspects ranging from the abstract why, to the concrete how representations of the product model. [Andreasen, 1991] [Krause et. al., 1993] [Ross and Schoman, 1977] [Ulrich and Eppinger, 2000] share this opinion on the different aspects over the life cycle. Further, it is also shown that these aspects can be grouped into four main categories:

- Requirements
- Functions
- Concepts
- Concrete solutions

Some researchers argue that a product can be described in more or less detail, by grading the description from low to high levels of detail.

In order to support different scopes and methods, different aspects of the product must be represented in the product model. These aspects can be represented in different types of product models and [Krause et. al., 1993] have identified five different types of product model:

- **Structure oriented product models** represent some aspects of a product from a product breakdown perspective. A product can be separated into functions and components. A typical functional breakdown is the Bill-of-Material and other assembly structures

- **Geometry oriented product models** represent the shape of the product which can be described in many different ways depending on the purpose of the description. Common model descriptive methods are wire-frame, surface, solid and hybrid models.
- **Feature oriented product models** represent the products in features. Furthermore, a feature consists of two parts, (i) a part that is application independent, (ii) a part that is application dependent. It provides the possibility of using different engineering applications and shares application independent information and customises it for the particular use by adding application dependent semantics.
- **Knowledge based product models** accumulate knowledge about the product. The accumulated knowledge is to be used as a guide to set constraints in the design space [Krause et. al., 1993]. Further, a knowledge based product model is mainly used for guiding and controlling designers when using other types of product models, rather than representing the product itself.
- **Integrated product models** represent a product by combining several different types of product models [Krause et. al., 1993].

Furthermore, product data management should not be confused with *product management*. The latter term includes business aspects, such as marketing, product policy and the introduction of new products, which is not included in this information domain.

5.2.2 Process Information Domain

The process domain or manufacturing process domain as it is also called represents all the relevant information about manufacturing processes throughout their entire life cycles. In this thesis a manufacturing process is considered to be a process which transforms a physical object from one state to another.

A process could be defined as a set of interrelated activities that transforms input to output, or in other words, a transformation process [Hitomi , 1979] and [Hubka and Eder, 1988]. Since the process requires a direction, declaring from which, to which state something is transformed, a goal orientation was added [Juran, 1988]. With this amendment to the definition, the process can be seen as a set of actions aligned towards the fulfilment of a goal. Further, it can be concluded that information necessary in order to describe the transformation from one state to another should be represented by the manufacturing process model. This includes information that specifies operations, resources and skills. This view is also shared by [Hubka and Eder, 1988], [Johansson, 2001], [Nielsen and Kjellberg, 2000] and [Tönschoff and Zwick, 1998]. However there are some differences of opinion with regards to whether the manufacturing resources are represented by the manufacturing processes or not. It can be noted that development has been toward describing manufacturing resources in a separate information domain, completing the PPR domain separation [Hubka and Eder, 1988], [Johansson, 2001], [Nielsen and Kjellberg, 2000] and [Tönschoff and Zwick, 1998].

The process information has been separated in three types of information that relate to each other [Tönschoff and Zwick, 1988], [Hubka and Eder, 1988], [Johansson, 2001] and [Nielsen and Kjellberg, 2000] [Nielsen, 2003]:

- **Process structures** - structural relationships between processes, e.g. decomposition, alternative and parallel
- **Process sequences** - sequential relationships between processes
- **Process parameters** - parameters and properties of processes, e.g. process time, process temperature, and feed rate

Furthermore, process information should also be product independent. This means that product information does not have to change to fit the process information. Through the complete separation of product and process information, reuse of product and process information has been enabled. For the same reason it is important to separate the resource information from the process descriptions and allocate it within its own domain.

5.2.3 Resource Information Domain

As previous indicated, a manufacturing resource model should represent all the relevant information about manufacturing resources throughout their entire life cycle. The information that constitutes a resource is often the same as for a product [Johansson, 2001]. Further, [Johansson, 2001] also means that the same information model can be used for the development of both manufacturing resources and products. This view is also found in [Hubka and Eder, 1988] who do not draw a line between different technical systems. For example, a car can be seen as a product, as it mostly are, however, it is also a technical system which is also a machine tool and a machine tool is mainly regarded as a resource.

Furthermore, both the car and the machine tool can be considered both a product and a resource. It can be said that if a product is not needed i.e. not useful for someone, it will not be a successful product. Therefore it seems the difference between a product and a resource could be an opinion. The same object can be a product to some but a resource to others.

In contrast to the information considered important from the product perspective, the important information from the resource perspective supports decisions concerning the development and realisation phase of an object. The resource information is also used to support different decisions concerning matters such as operation and support of the object, including capabilities, performance and operative expenses.

Resources and their capabilities

[Juran, 1988] has defined capability as the inherent ability to deliver performance. Interesting to note is the distinction between performance and capability. Capability indicates what a resource could do, whereas performance implies what a resource did do.

There are usually two types of capabilities discussed in literature, process and machine capability. Sometimes they refer to the same capability and sometimes they refer to different ones. The confusion can sometimes be big and the source of the confusion originates from a careless use of the term process for both manufacturing processes and manufacturing resources. [Bergman and Klefsjö, 1998] and [Juran and Gryna, 1988] do a distinct difference between the two but [Curtis, 1988] does not.

[Juran and Gryna ,1988] and [Bergman and Klefsjö, 1998] separates the capabilities in process and resource (Machine) in the following two definitions:

- Machine capability refers to the reproducibility under one set of process conditions
- Process capability refers to the reproducibility over a long period of time with normal changes in works, material and other process conditions.

This indicates that both processes and machines (resources) can have capabilities. However, in this thesis resources are considered to be the only one to have capabilities on their own. Further, capabilities that are modelled with processes need a resource to activate and carry out the capability.

5.3 Reuse of Information

Information systems are created with the intention of storing information so that it can be used at a later time. In relation to virtual manufacturing activities this has been utilised since the introduction of specialised information systems, e.g. PDM, ERP, etc. Traditionally the product side has led development and therefore IT systems are highly customised for the type of virtual manufacturing activities performed in this area.

When looking at the VM tools for production, e.g. discrete event simulation, there are very little done to support reuse of information. But the increased use of DES and CAR tools has lead to a new demand on the information systems, e.g. the development of MDM systems, section 3.2.2.

Traditionally, the information was accumulated during a project and used separately in each phase. Very limited parts of the information were passed on to the next phase. This was mainly due to two reasons:

- There was a change of personnel between project phases. Few people other than the project leader followed a project from start to finish. It meant that only a fraction of the experience from one project phase could be transferred to the next. It also had the effect that sub-optimisation are being rewarded because of the narrow focus of each phase.

- Coordination between two following phases was poorly managed. This means people working in one phase did not communicate with those working in the next. It had the effect that the same information could and had to be re-created several times during a project. The redundant work this meant was also a source of potential errors, which could easily affect the outcome of the total project.

In order to help cover the gap which currently exists between the different stages in the life cycle of information, it is anticipated by modelling the need of information in one stage that it will be easier to understand why certain information needs to be produced in an earlier phase. The bindings between the phases will also be more easily identified if proper information models are created.

Often during the identification and concept phases of a project only a few people are involved as the information starts to accumulate to a level required in the next phase. The information is continuously stored in the information system available. Several different computer tools are probably used and the information from them is stored without considering later reuse. When the following phase starts, much of the previously stored information is needed as a base for different decisions. It implies that data and information are accessible. This is true, but the systems software from the previously phase is very rarely used in this phase and much time is wasted on transforming the information to a readable status for the new software.

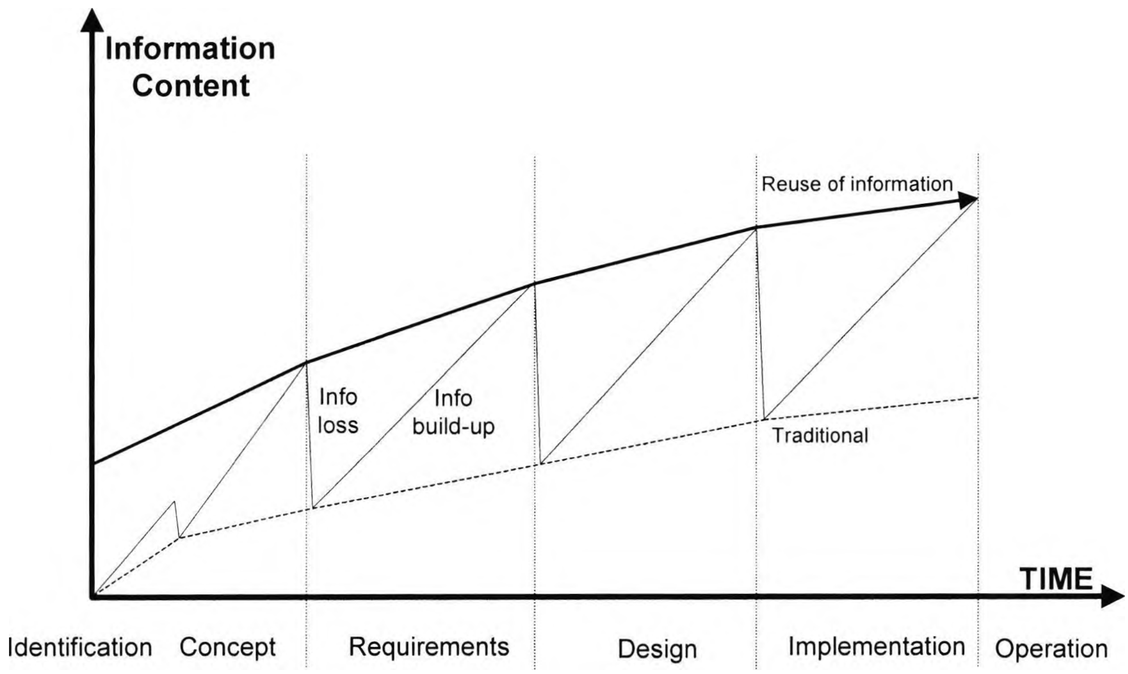

Figure 27 Reuse of information between project phases, modified from [Johansson, 2001]

With the use of a structured methodology and standardised information format it is possible to keep working even if software is changed or if the information has to be presented structured in a different way. Figure 27 illustrates the difference in information loss depending if the information can be reused or has to be recreated in every phase. Both time and resources can be saved when a larger portion of already created data can be reused, especially when it also contains structure from a standard and stored in a platform and application independent format.

5.4 Summary on Reusing Information for Manufacturing Simulations

In order to use information together with simulation tools, four different approaches have been identified [Robertson and Perera, 2002]. The approaches to feed information to a simulation system are:

- Manual input
- Use of spreadsheets

- Use of intermediate database
- Full integration with the corporate business systems

A company that uses simulation normally starts with the manual input of information and then advances to using spreadsheets instead. However, using intermediate data bases or having full integration with the corporate business systems is rarely implemented since it requires much more from the information systems that provide the system both in terms of information structure and use.

To approach full integration of the information sources with the simulation system the concept of dividing the data into product, process and resource domains has been explored. Within each domain the focus is to provide structure, semantics and preserve context.

By providing information with context, well defined semantics and a structure suitable for computer interpretation, simulation systems can be integrated with their information sources and reuse information. The full and direct integration of corporate business systems with simulation systems may still be difficult to achieve. Nevertheless, a mixture of using intermediate databases and full integration is achievable and even desirable when it comes to information security.

6 A Framework for the Structured Reuse of Product and Process Data

To demonstrate the use and reuse of product and process data for a virtual manufacturing tool a framework has been formalised. The aim of the framework is to demonstrate that parts of standards such as ISO 10303 and ISO 8879 can be used with existing information systems and that a planned structure can provide a means of reusing data which will make virtual manufacturing more effective and less an expertise.

6.1 Framework Development Process

The development of the framework has been case driven and it follows the qualitative study methodology. The literature survey served as a basis for this study, it identified a need, data and data sources, and a set of requirements. The development of the framework started with the database part and with the process of finding a data structure that could be used with simulation data, Figure 28. As this was an iterative process, several attempts were made to find a structure that could work. In order to be able to test the structure with the preserved aim of the data and models, some special attention was spent on the connection between the simulation software and the created data structure. This was the concern of the first experiment.

When a data structure that could handle the data was found, the second phase of the framework development started. This phase included the development of the user interface and functionality that would make the framework usable in an industrial environment. In the second test case, which was industrial based, the whole framework was tested. During the setup of the test case some functionality of the framework was changed as well as parts of the user interface, in order to harmonise with other systems used in industry.

As a final step in the development process the evaluation of the framework was done as part of the second test case.

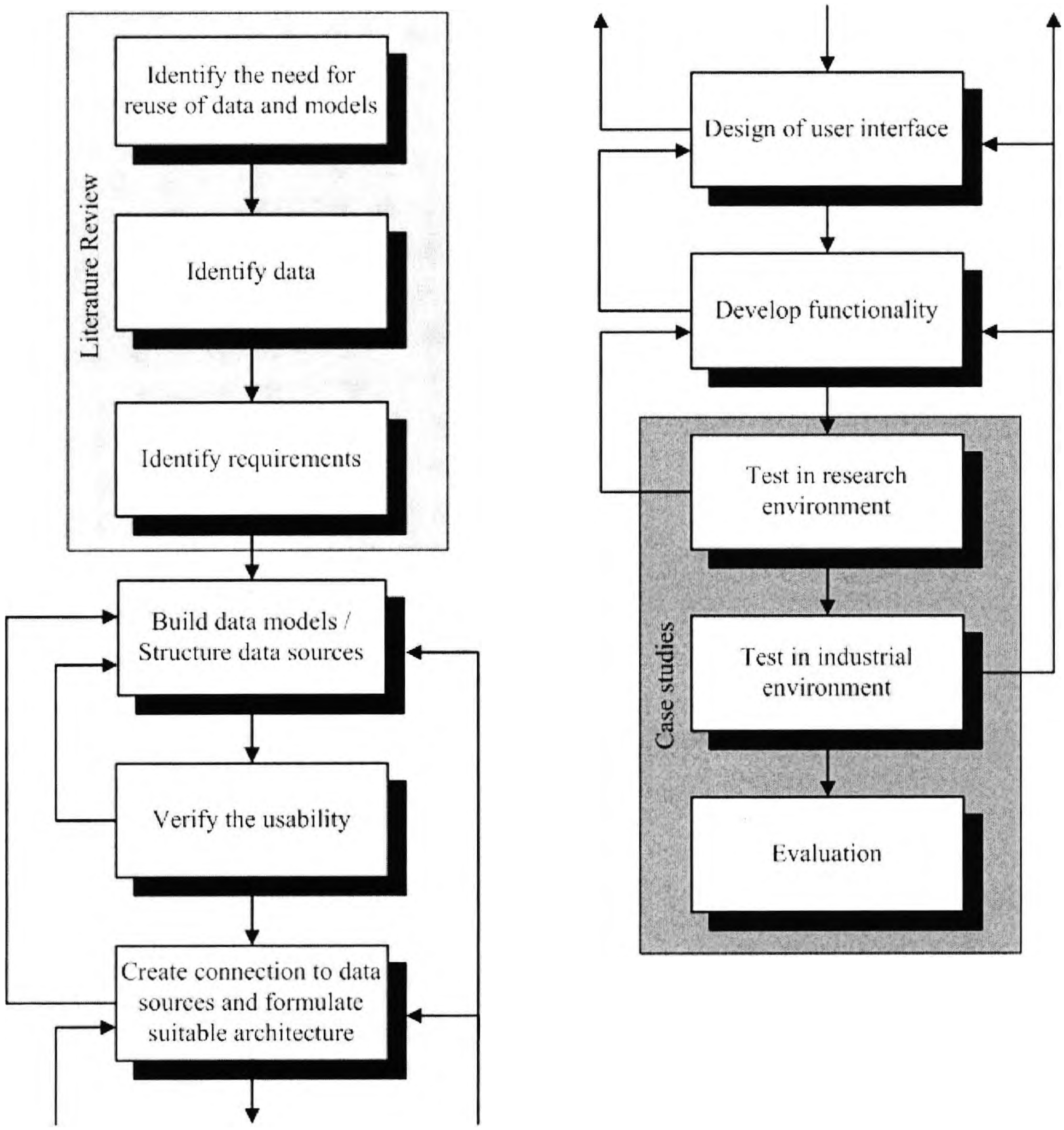

Figure 28 Steps in the formulation of the framework

6.2 Framework Overview

The framework that was formalised has a three-tier-architecture in order to achieve the best possible integration of the information sources with the virtual manufacturing tools, see Figure 29. Every level in the architecture consists of several different combined techniques and tools which together have the advantage of fitting into an approach which is also supported by several different standards from the ISO.

- RMSD is an information integrator for reusable manufacturing simulation data
  - User interface
  - Virtual manufacturing tool
  - Mapping tables and XML technology
  - DM-systems and STEP information models
- Domain data bus
- Data sources include:
  - Static data sources
  - Dynamic data sources

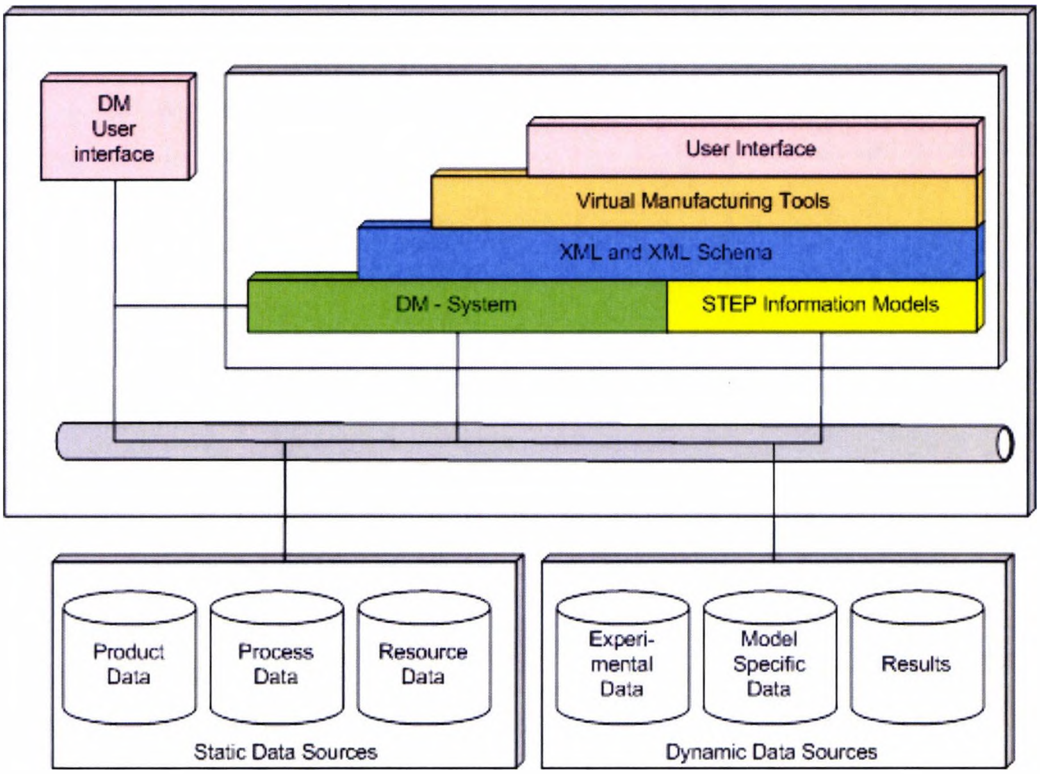

Figure 29 Information integration framework

6.3 RMSD

The information integrator system which is designed for reusing manufacturing simulation data provides a front-end interface for a user. The user interface should be similar to any other DM-system user interface on the market today in order for a user to recognise functionality.

In support of the user interface the virtual manufacturing tool has been integrated with data sources using standards such as STEP to provide the semantics for the data used and XML to provide a structure which is easy and robust enough to use in an industrial environment where huge amounts of data are required to perform a study.

6.3.1 User Interface

The user interface consists of three separate parts which together will provide the user with the information required to conduct a simulation study. The concept consists of a frame that encloses the three main information areas. The frame controls which model to work on, and the user can select freely from a list of models which are put into the system. By selecting one of the models that is in the system, the model is loaded and the PPR data for the model is shown to the left, see Figure 30.

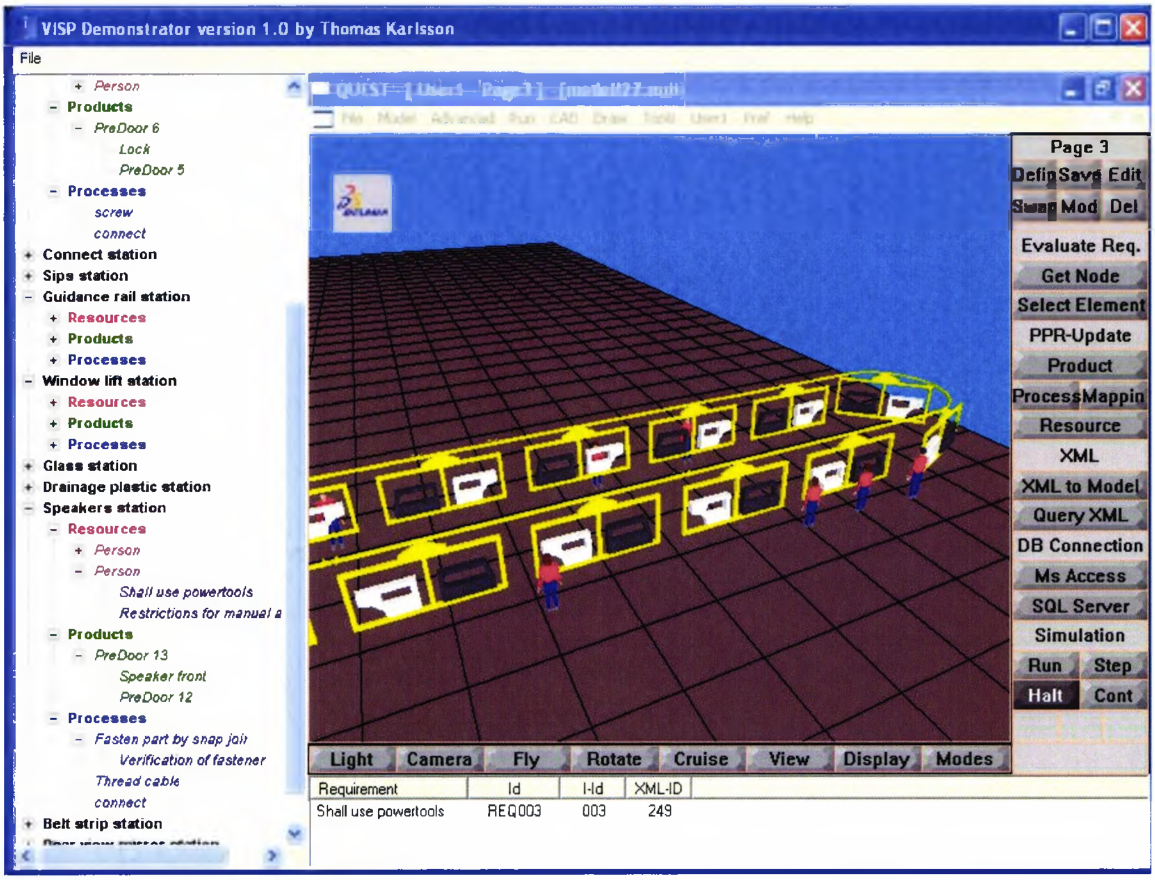

Figure 30 User interface with a loaded model

All the PPR data is displayed on the left of Figure 30. It is organised into the workstations that correspond to the simulation model. A station may include more than one resource, handle more than one product and be able to execute more than one process.

By representing the station as a node, the user has the possibility of expanding the node and viewing the number of PPR items connected to it. In the node, information that belongs to resources is represented as red, product information as green and processes as blue. This helps the user work with the information and minimises misinterpretations of the information which is connected to the simulation model.

Furthermore, there are often requirements associated with PPR information, which are important to acknowledge when a simulation model is used to test new scenarios or remodelled. Typical requirements represented in the PPR tree are those concerned with resources, e.g. to use certain power tools for assembling even when several others are available.

The requirements have a limiting effect on the simulation model and also on the designing options that are available when working with the simulation model. However, the representation of requirements together with the PPR information provide a structure to the requirements belong to a certain type of resource. Requirement modelling is also one of the areas where STEP has its ambitions, which is demonstrated through the development of STEP AP233.

At the base of the user interface there is a section which displays detailed information about the active node. Such information is to be used by the simulation modeller. It provides a more detailed explanation of the selected item in the form of relationships for the information system and other PPR data. However, a normal user of the system rarely requires the information from this window.

The virtual manufacturing tool is incorporated in the middle of the user interface. When the user has selected the model to work on it is displayed using the simulation tool. The simulation software is then working as the executor of the model and displays the changes that were made to the PPR-data. Further, the PPR-tree which exhibits the PPR-data is linked to the simulation tool in a way that it enables active use of both; e.g. the elements in the simulation model can be found using the PPR-tree and vice versa.

Figure 32 New menu for the framework

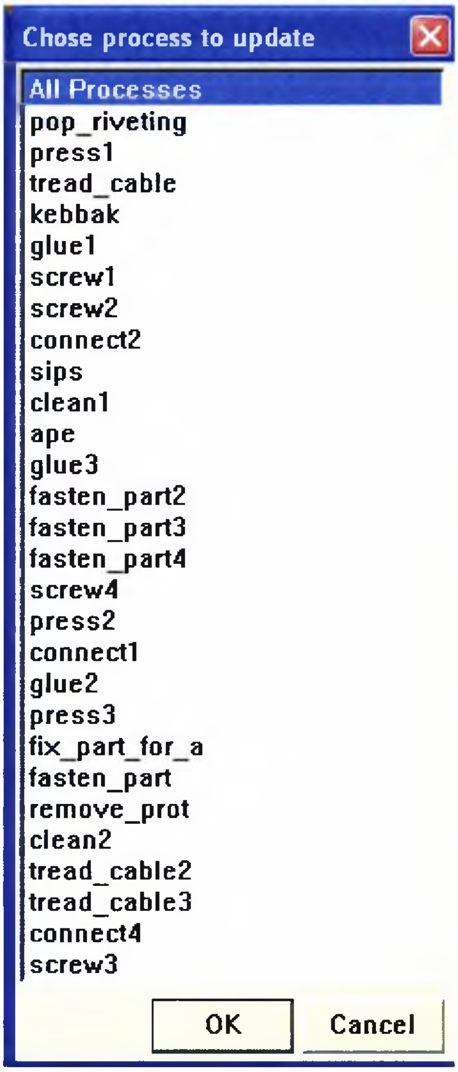

Figure 31 Process menu

The simulation software is supplied with a new set of buttons, which provide necessary new functionality for the framework to be operational, Figure 32. The buttons handle the communication with the underlying mechanisms that constitute the framework, i.e. data sources and mapping tables.

The functionality provided by the new sets of buttons includes:

- **Get Node** – The current active node in the PPR-tree is identified in the simulation model by a blue sphere placed above the resource or product. If a process is the current active node the resource that executes the process is indicated.
- **Select Element** – The last selected graphical element in the model is revealed in the PPR-tree which helps the user rapidly search and find the data required. The node is automatically expanded and activated.
- **Process** – By selecting the process button the user has the ability to update the simulation model with the latest process information. When the process button is selected, a list of all available processes that the resources in the model can execute is displayed. The user can choose to update only one of the processes or all. At the conclusion of the selection the OK button is pressed and the process data presented in the PPR-tree is read and implemented in the simulation model. When the model is run after this is completed, the new process data is executed.
- **Product** – Functions as process button but is used to update the information that describes the products the model handles
- **Resource** – Functions as process button but is used to update the information that concerns the resources.
- **Mapping** – Maps the process data to the resources, see section 6.3.3
- **XML to model** – Opens a XML parser which is used to read the PPR-data from the data sources. A local copy of the data is read into the RAM of the computer which enables rapid disconnected use of the data.
- **Query XML** – The user has the possibility to query the XML data which is placed in the RAM of the computer. XPath can be used to query the data.
- **Ms Access** – This is used to submit the simulation result to the information system. The data that is sent back to the information system varies depending on the

questions the simulation model is supposed to answer. MS access can serve as a database for testing and minor models.

- **SQL server** – Functions as the Ms Access connection but is supplied with the SQL server connection. The user will be prompted with server name, database, user name and password.

6.3.2 Virtual Manufacturing Tool

A discrete-event simulation tool called QUEST [QUEST 2005] is used in the RMSD. QUEST is a 3D- simulation package with state-of-the-art functionality. The package is widely used in industry, not only in the car manufacturing sector but also within ship building, mining and health care [Urenda et. al., 2004].

QUEST provides several means of controlling the simulation model for the user, who has access to a menu/button interface from which a model can be built and logically controlled. Furthermore, QUEST provides a powerful scripting language called Simulation Control Language (SCL). This is used to control the simulation model and to build mimics of the control logics which exist in a real manufacturing unit. SCL can also call C functions in a DLL, which provide full access to external software.

The simulation package QUEST also has a Batch Control Language (BCL) which can be considered a text-based interface to QUEST menus. BCL is mainly used to re-program the model which is useful when updating input data to the model. However, SCL and BCL together form a powerful method of extending QUEST for the necessary information sources.

6.3.3 Mapping Tables and XML Technology

The processes defined in the database describe how the manufacturing process is to be performed by the resources. The processes are defined to suit the products to be handled and the resources that execute them. Nevertheless, the link between resource and process is not defined by either the process or the resource in itself. The same applies to the relationship between product and process. Therefore, a mapping table has been developed in order to provide a context to the PPR data.

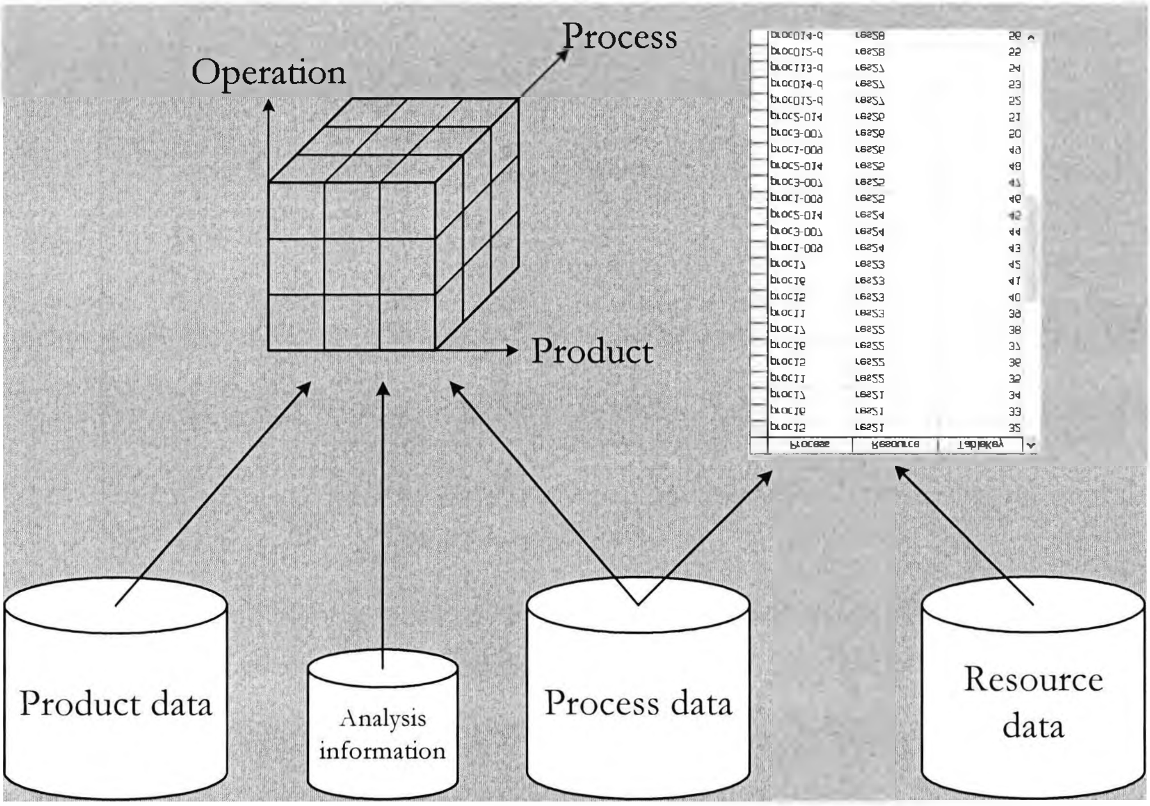

Figure 33 Putting PPR-data together

As indicated in Figure 33 the mapping is divided into two separate parts. First, there is the mapping of process data with analysis information and product data. This mapping is conducted in order to form the operation sequence for each product. In a simulation model, this type of mapping is most often coded directly into the model, using the knowledge from the production analysis department. Defining the operation sequence based on the processes can be very complex, especially if many different products use the same production facility and also the same machines. The use of PPR-data together with a mapping table provides the simulation model with the ability to define unique operations for every product in the system, even if a product is introduced to the simulation model at a later stage. By providing the product with its own list of operations which includes a number of processes, the simulation model is not required to have a hard coded operation list.

Secondly, the mapping of resource data with process data informs the simulation model which resources can perform a certain process, see Figure 34. It is common that several resources can perform the same process and that one resource can perform many different processes. However, restrictions may exist which indicate in what order processes can be performed in a resource. Those restrictions can be found in the resource data.

Process	Resource	TableKey
proc15	res21	32
proc16	res21	33
proc17	res21	34
proc11	res22	35
proc15	res22	36
proc16	res22	37
proc17	res22	38
proc11	res23	39
proc15	res23	40
proc16	res23	41
proc17	res23	42
proc1-009	res24	43
proc3-007	res24	44
proc2-014	res24	45
proc1-009	res25	46
proc3-007	res25	47
proc2-014	res25	48
proc1-009	res26	49
proc3-007	res26	50
proc2-014	res26	51
proc012-d	res27	52
proc014-d	res27	53
proc113-d	res27	54
proc012-d	res28	55
proc014-d	res28	56

Figure 34 Process – Resource mapping table

Using this mapping feature in order to change a simulation model is also suitable when newly designed processes are to be tested. The process is simply added to the process domain in the PPR-data and the mapping relation is defined in the mapping table. Updating the model is then a matter of clicking a button.

The basis for the functionality of the framework is the XML-format on which the PPR-data is communicated from the data sources via the data bus to the simulations system and back again. The use of protocol ISO 10303-28, STEP data to XML, has made it possible to extract information from data sources which has an information model based on STEP AP 214. Transforming the XML-data to suit the simulation model has been achieved using several XML related techniques. The data from the information sources includes much STEP specific information that belongs to the protocol. Therefore, the data has been cleansed to include only PPR-data, see Figure 35.

```
...
<Products id="001" version="1.0" name="Product domain">
  <generalproductrequirements id="k78" version="1.0" name="Common Product
   Requirements" description="" />
   <Product id="lf01" version="1.0" name="LFD" description="Left Front Door">
      <productrequirement id="rlf01" version="1.0" name="Fixing door to Carrier"
       description="How to fixate the door to a Carrier">
          <TextRep>The door shall be fixed</TextRep>
       </productrequirement>
   </Product>
   <Product id="lb01" version="1.0" name="LBD" description="Left Back Door">
      <productrequirement id="rlb01" version="1.0" name="Fixating door to
       Carrier" description="How to fixate the door to the Carrier">
          <TextRep>The Door shall be fitted to the fixture</TextRep>
      </productrequirement>
   </Product>
</Products>
...
```

Figure 35 PPR data in XML format

6.3.4 DM-Systems and STEP Information Models

Both commercial DM-systems and custom made information systems can be used in the framework which has been tested with two types of commercial software for static data management. They differ in the development and implementation of an information model. One system has an information model based on STEP AP 214 and the other has an information model not based on STEP AP 214 although there are functions that transform the output information into compliance with STEP AP 214 semantics.

The custom made information system is built with respect to the open PDM schema which includes a number of STEP protocols. Even though the system does not fully implement the standards, it uses the semantics and borrows the meanings of the features from the standard.

Therefore custom built information systems can be constructed and used with this framework, as long as they use the semantics of STEP.

The framework uses the semantics of STEP AP 214 in the whole chain, i.e., from the databases through to the simulation software. The semantics is also used when data is sent back to the information system.

6.4 Domain Data Bus

The domain data bus concept handles the connection between the simulation databases and the rest of the framework. It provides a link over intranet/internet which enables a distributed implementation.

In order to provide this functionality the .NET programming environment was used. The tools that are incorporated within this environment which is provided by Microsoft, enables integration with the databases while still managing the virtual manufacturing tool. ADO.NET which includes tools and functionality to handle databases contains two methodologies, namely connected manipulation of the data base and disconnected manipulation of the database.

When working with simulation data over a distributed environment, disconnected data sources make a perfect match. This is due to the fact that large amounts of data which might be required and data distributed continuously over a network could be very costly in terms of performance. Disconnected database connections mean that the client only connects to the database at certain times, which minimises traffic on the network. The data is read into the RAM of the client computer and manipulations to the data are conducted in this *local* copy of the data. When all manipulations are completed, the connection to the database is brought back up and the data is pushed back.

However, it is a disadvantage that a change to the data in the database after the data has been read does not show in the client copy of the data. Thus the client could be working on an obsolete set of data.

Furthermore, the data conveyed by the domain data bus is encapsulated in the XML-format which will ensure that the structure of the data is preserved during the distribution. The tags used in the XML-documents are decided by STEP AP 214 semantics which will ensure that the semantics of the data is compliant with the rest of the framework.

6.5 Data Sources

A methodological approach is required to identify, collect, organise and validate input data. Perhaps the best way to help the modeller to find the correct data in a structured manner is to build a customised information system for every single enterprise. As indicated, there is a demand for a computerised environment that contains a standardised representation of product, process, and manufacturing resources information. By using databases with a standardised representation of this data, the implementation and integration with virtual manufacturing tools becomes easier. Separating the data into different categories also facilitates reuse of the simulation model, as well as reconfiguration of the model.

6.5.1 Database Requirements

The developed simulation databases structure is created in such a way that it should satisfy all of the requirements of an information system in a manufacturing company as presented chapter 4. It has a clear definition over what kind of information should be stored, and the data is easily accessed. The created databases also provide a solution to the problems and requirements concerning product data and product data management. The database supplies the modeller with the correct format of the data in an easy, accurate and timely manner.

The databases can share information with other systems, such as spreadsheet applications and other databases, while integration with an arbitrary simulation tool has been used to be a problem. Most simulation tools of today require program unique tools for integrating the simulation tool with other applications. The use of the XML format provides such a program unique solution to the problem. The database also supplies the simulation modeller with a common definition and semantic of the information, which is made possible by using input masks in the database.

By selecting a common DBMS such as SQL server or Oracle, several advantages can be reached. Users of the simulation database will quickly become familiar with its well-known working environment. It facilitates the exchange of data with other types of databases and by using standard tools it ensures that the concept will be in use for a longer period of time. All this together secures the data portability and longevity requirements placed on the system.

6.5.2 Static Data Sources

A database management system has a number of advantages, and can meet many of the demands on how to store, manipulate, and manage input data. The database can store the data required to define all manufacturing activities at the desired level of detail and can also retrieve and manipulate data according to the users' requirements by using a query language such as SQL.

The work of creating the database begins with analysing the requirements for input data in the virtual manufacturing tool used in the framework. After identifying the requirements input, it was classified into one of the three categories: Product, Process, and Resource (PPR) [Bernard, 2001], [Liyanage, 1999] and [Bacha and Yannou, 2002c]. It can be concluded that the categorisation of the static data can be conducted according to the following:

Product data: Includes all objects relating to the product; geometry, dimensional data, generic parts, sub assembly, variants, function etc.

Process data: Contains all the types of operations (manual or automatic), their description of assembly operation, geometry, cycle time data, etc.

Resource data: Covers the layout, human resources, machines and tools assigned to the station, process resources and requirements.

Dividing the manufacturing data for building simulation models, into only three categories can result in three very large classifications of data. This may result in data that is difficult to work with, and delays in finding the correct data. However, most simulation packages divide data into special element classes such as, Buffer, AGV, Machine, Source, Sink, etc. This implies that the static data in the database is not required to be classified into more than the three previously mentioned categories/domains (PPR). The data in each domain is consequently separated and assigned to a special element class which matches the one found in the virtual manufacturing tool. This means that the manufacturing data is first assigned to a special element class and then categorised into one of the PPR categories.

After reviewing the conceptual design of the database and evaluating query performance, the relational database is created using the defined entities and attributes. However, as previously mentioned, all of the created tables and attributes may not be required in a given simulation project since their use depends on the purpose of the project, and the manufacturing system under investigation. Therefore, each company may require its own customised simulation database but would benefit from a common minimally agreed unit to provide scalability.

The developed ER-model for the simulation data is converted into a MS SQL server database. In the SQL server, which is a relational DBMS, each entity in the ER-chart becomes a relational table and each attribute becomes a column in the table. Primary and forging keys are declared for every table as well as referential integrity constraints for every relationship.

Testing several alternative ways of structuring the database, results in the described database structure. An alternative database structure is achieved by separating all the parts, one for every specific entity (i.e. Buffer, Machine, Source etc.). This will result in a larger database due to the increased number of data tables that have to be used. This version of the simulation database design has both advantages and disadvantages associated with its design.

The main advantage is that it is somewhat easier to find the data associated with a specific entity. On the other hand, the main disadvantage is that it creates problems when searching and retrieving data that concerns a complete manufacturing sequence. The sequence normally consists of several machines, buffers and material handling systems which have a relation to each other. Mainly due to its rapidly growing complexity as a simulation model becomes larger, structuring the database by separate entities was abandoned. A simple ER-chart for this database is shown in Figure 36.

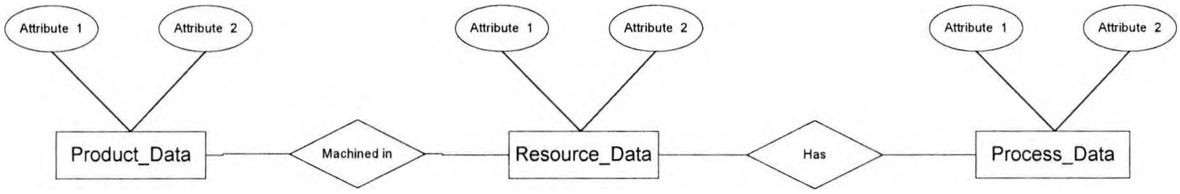

Figure 36 ER-chart for the bonded database design

6.5.3 Table Design

The specific simulation database is created according to the following. The *Product_Data* table contains all the products and specific data assigned to the product. In this table *Product_Name* and *Article_Number* are the primary keys as only one product can be assigned to a specific *Article_Number*. The *Available_Resources* table defines all of the available resources. In this table the attribute Available_Resources is assigned as a primary key, and every resource has a unique number. The *Product_Data* table and the *Available_Resources* table are connected by the *Product_ Operations* table. In this table the manufacturing of a product is assigned to a manufacturing resource. The manufacturing of a product can be assigned to several different resources, and the resources can be assigned to the manufacturing of several different products.

In the *Available_Resources* table, the different entities of resources, (e.g. Machine, Buffer, Source) are assigned to a unique range of "id numbers". The different entities of resources are then linked to the entities unique *Resource_Data* table, in which the data of a specific resource is assigned. The *Resource_Data* table is then linked to the entity unique *Process_Data* table. A resource can be assigned to several different Processes and different versions of the process.

Adding more resources, processes or products to the database model is normally achieved by downloading them from previous simulation models using the framework or using the ERP and PDM system. The latter is made possible through a direct user interface that is connected both to the DM-systems and the domain data bus. For instance, if a user would like to add Sink_Resource_Data, the modeller just adds the new element class to the Available_Resources list. After entering the new element class, the new objects Sink_Resource_Data and Sink_Process_Data need to be created. This is best achieved by modifying one of the already existing resource objects and process objects from one of the existing element classes, saving a new resource and process object and then changing the attributes that differ from the parent object. After customising the new objects, they just need to be linked to the main database model and if there are other relating processes those connections must be specified.

The user interface, which is connected to the DM-system, and the domain data bus can also be used for retrieving data from the system to a simulation model. However, doing so would mean a manual interpretation of the data and also result in non-use of the framework. The functionality implemented in this user interface is based on a script language, which creates detailed SQL queries based on the users' interaction with the system. The simulation modeller can obtain specific data that is desired in the simulation model from this user interface, but he is advised to only use this possibility under certain circumstances. Figure 37 shows a query form for Machine Resource Data.

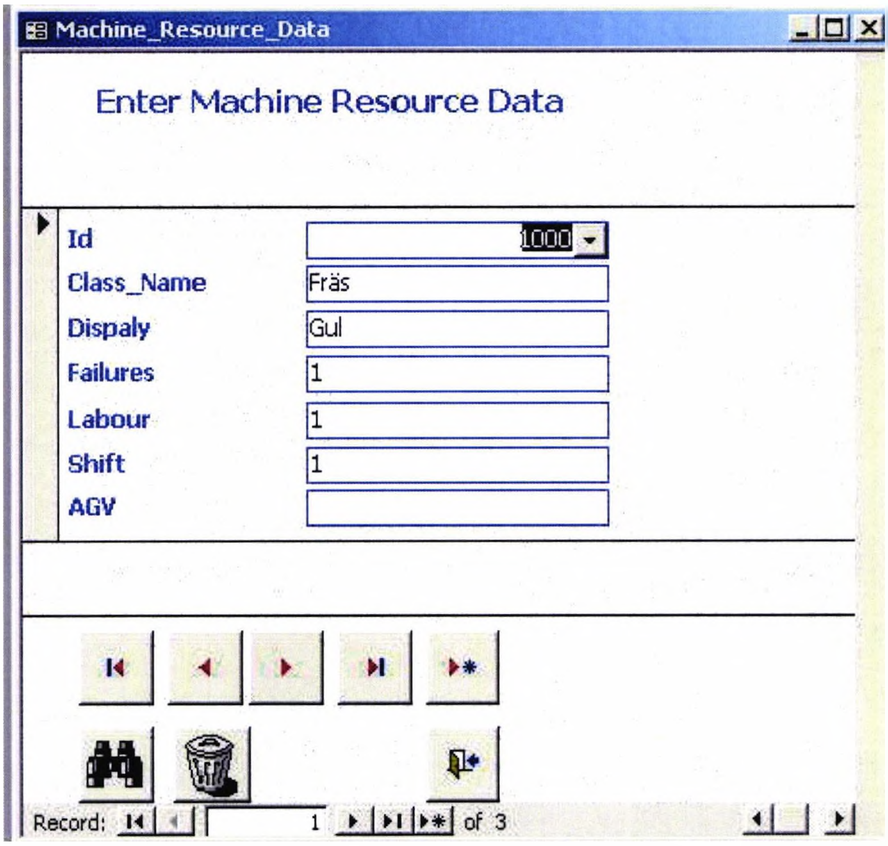

Figure 37 Machine resource data form

In most cases the simulation modeller is interested in the latest version of the data. Therefore, one of the most important features in the implementation is being able to select the desired version of the data.

6.5.4 Dynamic Data Sources

When examining the PPR concept in greater detail and attempting to use it as a solid information platform that can support discrete event simulation in a way that both makes simulation modelling faster and easier and information sharing natural, it can be seen that the concept has limitations. As stated by [Johansson, 2001] and [Bernard, 2000] PPR domains support information that can be clearly classified into either one out of these three domains. Another contribution of information modelling using the PPR approach is presented by [Nielsen, 2003], and shows that following the same approach as previous research initiatives even the manufacturing processes can be captured using the ISO10303.

However, not all the information that is required for conducting a discrete event simulation study can be categorised into these domains. Typically this means that more information can be found in the traditional information sources for virtual manufacturing, which do not fit into the PPR definition. This also suggests that the support from well-structured information sources based on ISO10303 is unlikely to provide all the information needed.

Further, the data to be considered as a candidate for the dynamic data domain has clear connections to Product, Process and Resource data but cannot distinctly be placed into one of these domains. The typical data in this domain is likely to change during a simulation or a set of simulations, therefore it is labelled dynamic data.

6.5.4.1 Experimental Data

The process of developing a simulation model is extensive in many ways. When the development is more or less complete the aim of the model is to help the user understand the characteristics of a certain pre-formulated problem. It is not uncommon that the nature of the problem is to evaluate several different configurations of the system under the simulation study.

Typical data comprising the domain of experimental data:

- Simulation time
- Dispatching rules
- Product mix

In the framework, the experimental data is used continuously when the simulation model is run. Experimental data is mostly decided in advance and limited to a certain range for a parameter.

6.5.4.2 Result Data

The result data is linked to the experimental data in a way that the two could form a circle of input-output data. One simulation's result could serve as the experimental data in another simulation. This indicates a very clear connection between the two data domains for a simulation.

Typical data that is included in the result data domain includes:

- Utilization
- Work in Progress
- Downtime

Output data or result data varies substantially from model to model. No general guidelines can be provided in the kind of output information that is of interest from a simulation study. In this framework, the most common parameters are automatically identified as output and sent back to the information system as a result. However, an analysis of the information that is put into the system is outside the scope of this framework. Nevertheless the output data for a certain model configuration can be found in the system and used as input parameters for an output analysis of the simulation result. The data that is sent back to the system uses the semantics of STEP as far as possible.

6.5.4.3 Model Specifics

The data domain that includes the model specific data is the smallest one. Only data that does not fit into any other domain is included here. The characteristics of this data are closely tied to one model and cannot be reused for another. However, some smaller parts of the data could with some minor modifications serve as inspiration when developing new models.

Typical data that is classified as belonging to the model specific data domain includes:

- Model Logics
- Model description
- Original system and version

The data in this domain is specific because of the way simulation software is built and executed today. Control logics and model descriptions rarely follow any major standards.

Instead, they tend to use the standards to develop their own languages and descriptions, which makes it very difficult to transfer one model from one type of platform to another. The framework could be ignoring this type of data and still be successful in the other five domains. On the other hand, the framework would then not acknowledge the fact that this type of data exists and is creating obstacles to the development of reusable data and simulation models for virtual manufacturing.

6.6 Using the framework

When using the framework to update a model with a new set of information, it provides functionality which makes this process much faster that before. Since the framework provides a hierarchal visualisation of the data that is included in a model, it is easy for a user to search and identify one specific parameter. When the parameter is found the framework could be used to change the value or the use of the parameter.

Action	Steps using the framework	Steps not using the framework
Change value on one parameter	*1. Select the parameter in the simulation tree* *2. Select Update button which downloads the new parameter value automatically from the information system in right format and compiles the code* *3. Run the model*	*1. Open all processes until the parameter is found* *2. Find the parameter in simulation code* *3. Search the information system for the new value* *4. Calculate value if needed and transfer the new value to the simulation code* *5. Compile and debug the code* *6. Run the model*

Table 1 An example of the different steps when using the framework and when not

Before the framework was formalised, a change to one specific parameter in a simulation model could require expert knowledge of the simulation software, programming and where to find the data in an information system. The number of steps was reduced when using the framework, see Table 1. Also the time for making a change was drastically reduced from as much as several hours to a couple of minutes or less.

6.7 Summary on Framework Design

The designed framework for reusing data and models for simulation consists of three parts: (i) information integrator, which is comprises (a) user interface, (b) virtual manufacturing tools, (c) mapping tables and, (d) information models (ii) the Domain data bus, and (iii) Data sources.

The information integrator provides a user interface which gives the user information about the current configuration of the model and data. Incorporated in the user interface is also the virtual manufacturing tool which displays the model. Supporting the user interface, mapping tables are using XML technology to connect data in the data sources to objects in the simulation model.

The domain data bus provides the connections between the simulation databases and the rest of the framework. It has been formulated using Microsoft .NET and the modules available therein. The domain data bus enables disconnected database management which is very suitable when working over the intra/internet.

Data sources which constitute the base layer in the three-tier design of the framework has been divided into two groups: (i) static data source which includes (a) product data, (b) process data and (c) resource data, (ii) dynamic data source which includes (a) experimental data, (b) result data and (c) model specific data. The separation of the data into these two groups provides possibilities of extending the framework for other information sources which might be hosting one or several of the different sub-grouped data.

7 Validation of Research Concept

In order to help formulate the proposed framework, one experiment and one test case were conducted. The first experiment was selected with the aim of defining a database which could contain simulation data and be able to preserve the original context of the data. The experiment had also to include a study on how to use ISO standards together with simulation data. The industrial test case was selected to be conducted in an organisation which already uses simulation as an engineering tool and also got several information systems in which data is stored. The test case was selected with respect to the designed framework and the possibility to test the functionality of it. The case focused the usability of the framework for matching the requirements between products and production equipment.

7.1 Asynchronous Updatable Simulation Models

The experiment was conducted from June to August 2003 at the University of Skövde.

7.1.1 Overview

The background for this experiment is that the use of production simulation tools is becoming more extensive. With the increased us of production simulation comes more requirements of data management. It is not uncommon to test several different scenarios based on the properties of the system. By changing some of the inputs, which is data, it is possible to visualise and measure the impact of one parameter for the whole system.

The aim for the experiment was to see how simulation software could connect to a database and use the information. The connection between the simulation software and database system had to enable structure and semantic preservation of the data. The context had also to be maintained trough out all the data translations.

As commonly known, simulations are most often complex in their nature, requiring many different sets of data, usually from several different sources. This experiment was set up to use data separated into product, process and resource data which could be seen as different data sources with relations. Further, the experiment elaborated with the possibilities to use ISO standards including STEP to communicate data between the simulation model and database, in order to maintain context of the data.

7.1.2 Results

The experiment resulted in the creation of a database structure that supports the data that a simulation model requires. The relationship between the complexity of a simulation model and the database structure was studied. As expected before starting the experiment a relationship showing that increased complexity of a simulation model requires a larger database structure to maintain the data. When the relationship was identified, only a smaller model was tested together with the database in this phase, in order to maintain the objective of the experiment.

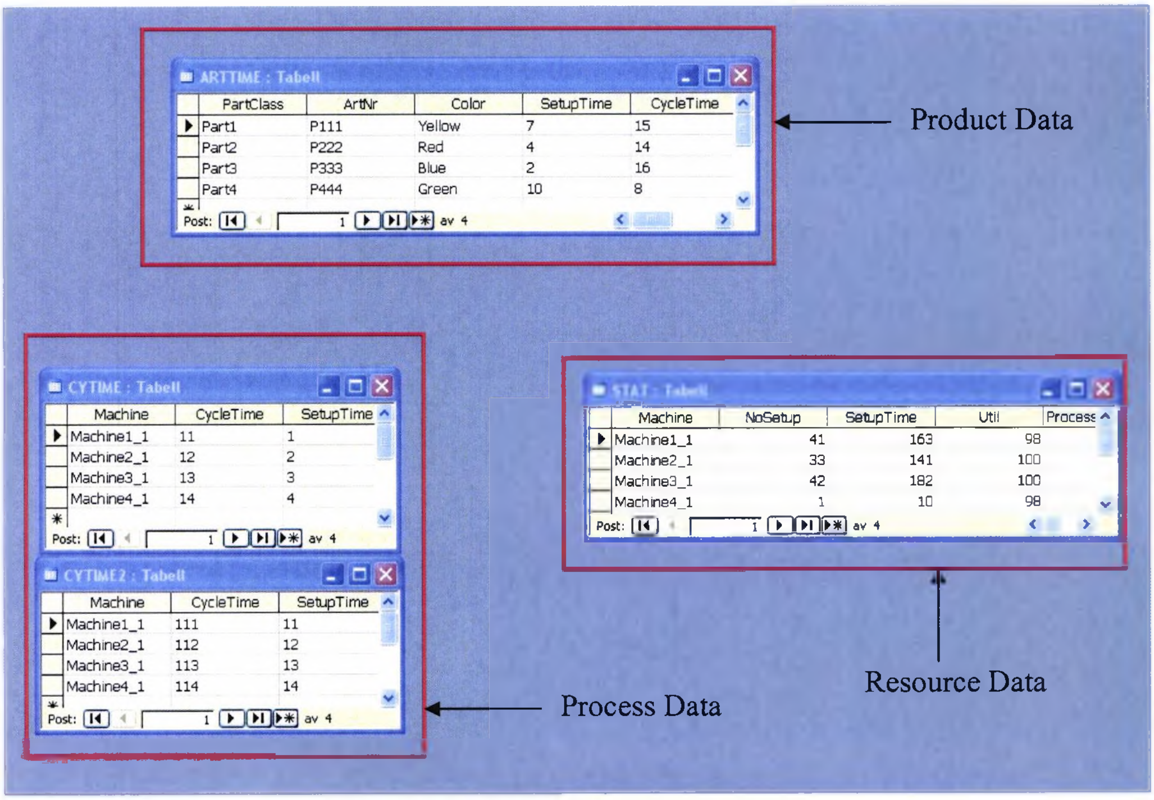

Figure 38 Database structure showing product, process and resource data

The database structure was modelled with respect to the separation of the data into; product, process and resource domains, as seen in Figure 38. This separation of data into domains provided the internal structure of the database.

Since it is specified how the three data domains interact in the simulation model, the data in the database can be maintained domain independently which also means that one domain could equal one database.

Further, when the database was modelled the data had to be reachable from the simulation model in order to be used as parameter data. In the experiment a small simulation model was created, representing a small manufacturing system, Figure 39.

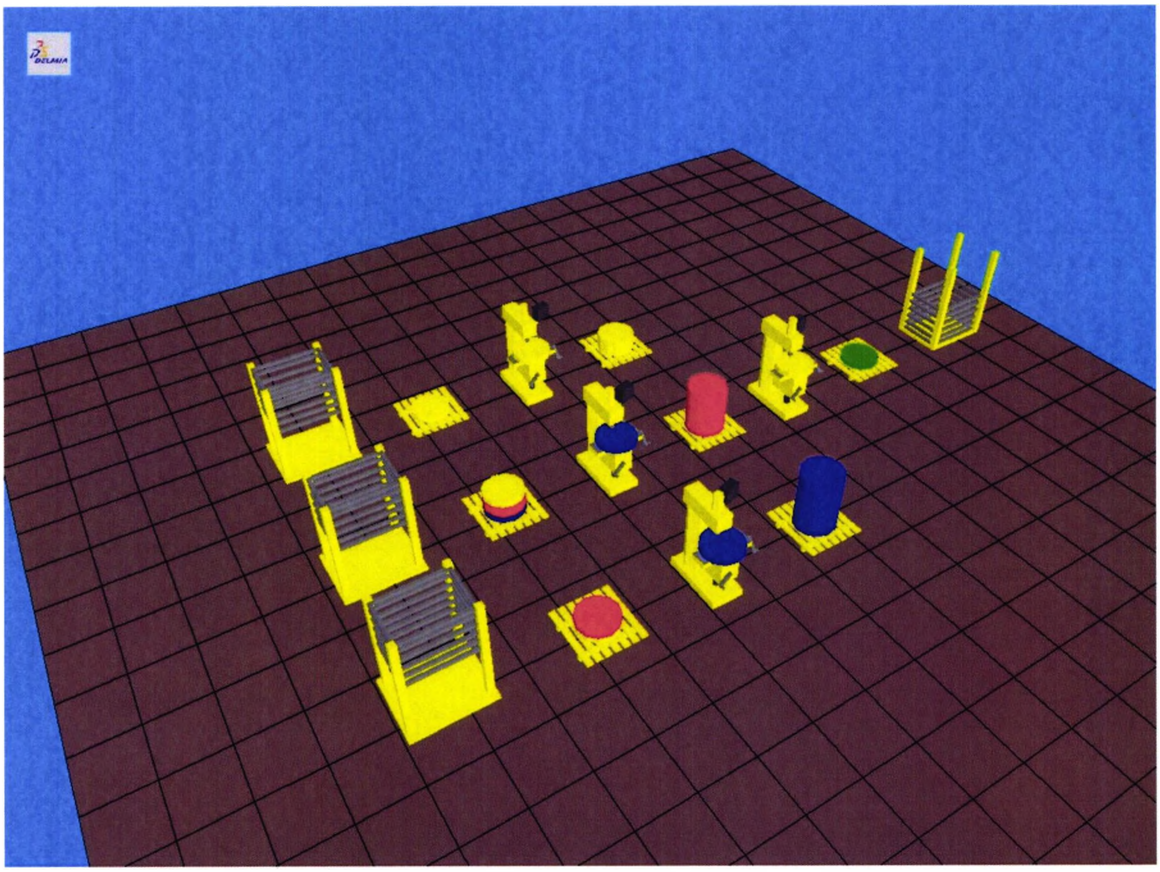

Figure 39 A simulation model of a small manufacturing system

This simulation model was connected to the designed database in order to test how the data in the database could be made available to the simulation model, preserving structure and semantics. The data is then used in the simulation model to make decisions when running the model.

The data to the simulation model was provided using the XML format which preserves the structure of the data from the database all the way to the simulation model. Further, the benefit of using XML documents as information carriers are that it has the possibility of include semantic support from other ISO standards. That possibility was utilised and the semantic from STEP was used to make sure that the data would be interpreted correctly when reaching the simulation model. The context support in this experiment consists of human interpretable notes that follow the data. By using these notes the simulation engineer has a possibility to understand the data better.

When testing the simulation model it could receive data from the database and updated product, process and resource data independently of each other. This provides the modeller with the possibility of easily updating the model with the latest configuration of, for example a changed process.

7.2 Virtual Integrated Simulation in Product Realisation

The goal of the project was to provide support to the product development through the use of simulation. It also aimed at making simulation easier to use and a natural tool to use in the development of new products. The VISP project was a co-operation between 3 different universities: the University of Skövde (HS), the University of Linköping (LiTH), and the Royal Institute of Technology (KTH). One research institute and three companies were also involved: Industrial Research and Development Corporation (IVF), and three companies: Volvo Cars Corporations (VCC), Swedish Ball-bearing Company (SKF) and Volvo Construction Equipment (VCE).

7.2.1 Overview

The background for this case was that products of today have a much shorter life cycle than the production systems which manufacture them. It has therefore become vitally important that new products are able to use existing production system processes and resources with minimal reconfiguration. The complexity also increases as the product design has to consider the manufacturing capabilities and the products currently manufactured in the system.

The aim of the case is to see how DES models could be used when introducing new products to an existing manufacturing system with existing production. The information that simulation tools could provide can help to identify areas in a manufacturing system that can cause problems when running a manufacturing system. The purpose is to help allocate resources to the sections in the production system that must be rebuilt in order to manufacture the new product and avoid spending time and money on sections which need not be changed.

Data that is needed by a DES model to provide a result is kept in numerous information systems in a company. Therefore must the information sources be identified and made available to the simulation tool. This could be done using a framework such as RMDS, which provide the necessary infrastructure and connections. DES model could then be the tool that could give answer to many of the questions which arise when introducing a new product to an existing production system.

7.2.2 Results

Prior to starting the case, the information necessary for decision making was dispersed in a number of different systems, see Table 2. It was therefore difficult to find the information that was needed for building and running a simulation model. The utilisation of an information framework, which could integrate with the information sources, ensures that required information could be found with less effort than before.

PDM system	Product structure, Product descriptions, Product geometry
ERP system	Cycle times, MTBF, MTTR, Product mix, Production batch size, Staffing
Requirement System	Product requirements on the production system, Interrelation of product and process data
Simulation Database	Simulation model, Simulation logic Simulation scripts

Table 2 Simulation data found in different information systems

The information framework which was developed during this test case had to be able to represent all the information to the user in a manner that could be easy controlled. All information was therefore represented in a hierarchical structure based on the manufacturing system structure.

This framework for using simulation data and models was also used as a simulation platform. The basic concept of the simulation platform, as discussed previously in this thesis, is that the data and information required to create one specific model can be reused at a later point in time.

In this case study it was required to create a simulation model based on an existing manufacturing system. The simulation model should also have the possibility of being used with the information sources and the requirement information platform. The developed

RMDS provided the necessary architecture. The identified data was linked to the two data sources included in the framework and a simulation model was built.

The simulation model built in the case study represents an assembly line where car doors are assembled by humans using hand held tools, see Figure 40. The simulation model is modelled on station level. It includes all resources in terms of humans and tools for assembling the products.

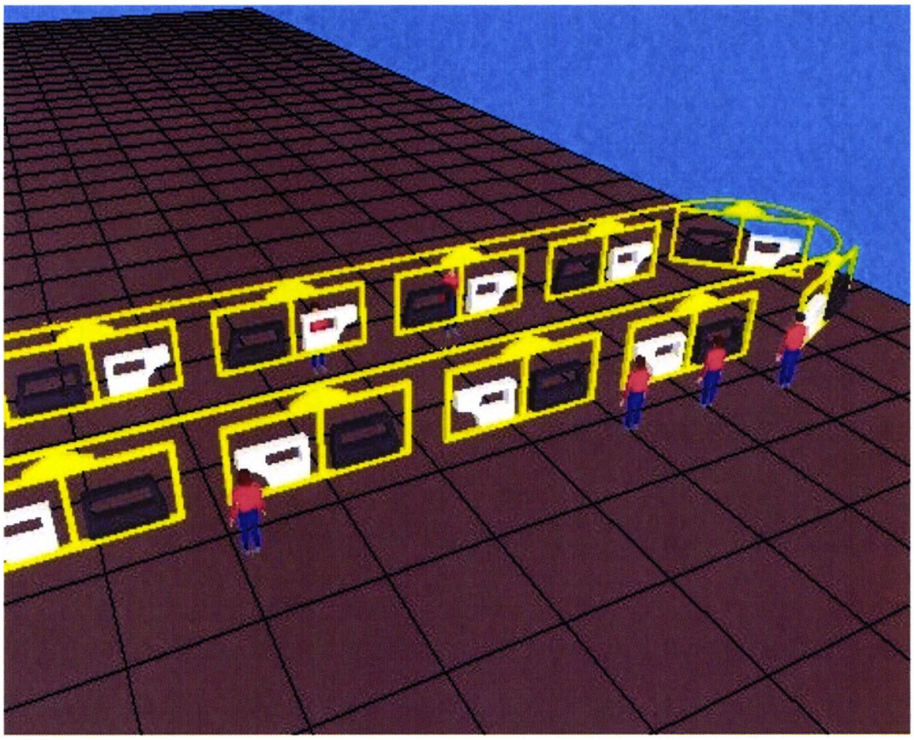

Figure 40 Door assembly simulation

Further, the sub flow that supports the main flow was not included in the model as well as main flow before and after door assembly line. Data used for running this model on a selected detail level includes quantified product, process and resource parameters which were collected from the identified information systems.

During the development of the simulation model, the logic of the model was designed in a way which allows several of the properties of the components to change over time. This means that the data used by the simulation model during a simulation run is transferred from the information systems and other legacy systems to the model when the simulation is initiated. Consequently, the result from an ongoing simulation could in some cases be considered to form input data and part of the requirement specification for the specific simulation task. When a simulation model is "feeding itself" during a simulation run, it was found that it is important to:

- *Build the model for data reuse*: acknowledge the relationship between products, processes and resources making it possible for their introduction to the model at a later stage.
- *Select level of detail*: ensure the possibility of configuring the simulation model before running, e.g. abstraction level, section, question to investigate.
- *Provide access to necessary information sources*: in a simulation study, access to sufficient and adequate information for the model is crucial for success. Thus the use of a common information platform should be standard.

As the aim of this test case was to investigate how DES models could be used under introduction of new products in an existing manufacturing system with existing production, a new door type was presented to the simulated manufacturing system. When running the framework with the new product it was possible to note a number of interesting problems with the manufacturing. The framework successfully identified two missing processes that were required for the new product, one resource shortage as well as a product mix problem. To overcome the identified problems, built in functionality in the framework was used.

Successful changes to the model using the framework:

- Re-arrangements of the process structure
- Change of a process parameter value such as process time
- Change of available processes for a resource
- Introduction of a new product into an existing production line
- Insertion of an extra resource in one station

From the case it could be concluded that the framework could be used to identify problems that present it self when introducing a new product to an existing manufacturing system. The framework also helps rearrange processes and resources within given limitations for solving manufacturing system design related issues.

However, some limitations were noted when the framework was tested. These limitations originate from the design intent of the framework. The framework is not suited to create new models based on none existing data, it is designed to collects and reuses data form several information system. A second limitation are the impossibility to expand the simulation model with extra stations though drag and drop functionality in the station tree structure window, such amendments to the model has to be done using the simulation tool.

7.3 Summary on Case Findings

The two experiments conducted during this research had each a different aim but sharing part of the background. The first experiment was conducted during the formation of the framework. It provided vital input to the design of the data sources. From the first experiment it can be concluded that STEP and XML can be used with success when supporting simulation models with information. The experiment also demonstrates that the semantic support from STEP will help the application interpret the data.

The second experiment, which was conducted with data from an industrial environment, used the formulated framework to provide an environment where models and data could rapidly be changed to test several design scenarios. It was found in the experiment that the framework could be used to match the requirements from products with the capabilities of the manufacturing system. It can also be concluded from the experiment that reuse of data and models is possible using the result from the first experiment with off-the-shelf virtual manufacturing tools which are put into a framework.

8 Discussion, Conclusions and Future Work

The focus of the investigation conducted in this research is on the reuse of data and models for virtual manufacturing with a special interest in discrete event simulation. The thesis argues that reusing data and models can save time and money and thus potentially offers advantages when reconfiguring existing manufacturing equipment. The investigation addresses key issues such as minimising modelling time and data collection. For the modern manufacturing/engineering industry, the major benefit is the potential to rapidly identify which resources along a production line must be given new capabilities, in order to manufacture a new product. The conducted research and formulated framework provides a decision base for the customisation of existing equipment and processes.

8.1 Discussion

In literature it can be seen that virtual manufacturing has emerged as a key technology to support and improve manufacturing operations in the 21st century. It has been reported to offer a huge potential for improving products and manufacturing processes, reducing production-to-market time as well as costs. Furthermore, in the literature survey starting out this research it has been noted that simulation models are often developed for one-off use only, despite the cost and time required for their development. By utilising custom made models, the capability of representing more than one system is almost non existent. Therefore it has been difficult to reuse a whole or part of a model for a similar system with a slightly different context.

Literature shows that the engineering community has been investigating the possibility of standardising information representation and modelling for more than 20 years. This led to several initiatives from which STEP emerged as one of the most comprehensive and important standards. It is noted that STEP provides the engineering community with semantics for the data, which opens new possibilities for the development of integrated engineering environments. However, for efficient and effective reuse of data and models, virtual manufacturing systems and standards have to be combined. In order to achieve this, a number of requirements have been identified which must be fulfilled to build a functioning framework that enables reuse of data and models for discrete event simulation. These requirements have been classified into three groups:

1. *Simulation system requirements*
2. *Data requirements*
3. *Architectural requirements*

Based on the findings from the literature a framework was formulated. It was designed with a three-layer-architecture. The aim of the framework was to provide engineers with an integrated environment in which simulation data and models can be reused without reconstructing the models from scratch. The actual design of the framework was experimental driven but with a strong basis in the identified problem area. This gave the formulation process a bottom-up characteristic which began with the formulation of the data structures in relation to simulation models and ended with user interface design.

The theory elaborated within this thesis has prepared the framework for reuse of product, process and resource data in discrete event simulation models. The formulated framework also harmonises with international standards on information management and modelling, as it makes use of the identified standards in this research. These standards also provide semantics and structure to the data.

After arriving at a framework that encapsulated the research aim, the industrial test case was also used to validate the framework. It was done by using the framework to introduce a new product into an existing production line. It was tested if the framework could identify resource capabilities and product requirements, and potential mismatches, in an industrial environment. The framework successfully identified resources along the production line that did not have the required capability to process the product.

8.2 Conclusions

A data handling theory for virtual manufacturing has been presented in this thesis. It consists of a system domain specification, which identifies that data can be divided into product data, process data and resource data. It also defines the relationship between these data domains and supports it with structure, semantics, and context which enable data reuse in virtual manufacturing systems.

The major conclusions are:

- Virtual Manufacturing covers both product and manufacturing system development from early phases such as concepts to production and beyond. Every virtual manufacturing tool uses data and models that provide increasing knowledge about the products and production systems. The time and effort for creating these models can be reduced considerably if data and models are reused across the whole virtual manufacturing domain. A framework that adequately supports reuse of data and models thus facilitates both product and manufacturing system development
- Communicating, as well as and sharing data and information requires an agreed format. The benefits of using a format that is defined through ISO standards are several, e.g. good documentation of the format, the possibility of new tools using old data, and independence of an enterprise's internal standards. Therefore several ISO standards have been used during the research. It can be concluded from the result that a framework for reusing data and models that uses ISO standards is free of any enterprise or manufacturing domain and possesses a scalability
- A framework that is based on the identified requirements for information and model design and reuse has been formulated. It is also designed to enable distributed integration of information sources and the virtual manufacturing tool, in this case the discrete event simulation tool QUEST.
- Input and output data for discrete event simulation models have been identified and classified into six groups. The groups are used by the framework and categorised into dynamic and static data for the simulation model based on their use in models

- Two experiments were conducted during the research. The first experiment helped formulate the database and understand the relationship between different data. The second experiment illustrated how the framework could be used in an industrial environment and what types of issues could be addressed with the framework.

- This thesis has explored the information structure of the data required to conduct a simulation study for the possibility of reusing data and information. This suggests that data and information from previous studies can be reused, and likewise, future studies can utilise data and information from current studies. The framework proposed in this research increases the reusability of the data.

8.3 Contribution to Knowledge

Specific scientific contributions to knowledge for this research study include:

- The research successfully demonstrates how theories and techniques in the area of information management can be applied to virtual manufacturing. It establishes that a virtual manufacturing tool can be used in an information and knowledge driven environment, taking into account the application context, data structure and semantics

- A novel technique for integrating simulation software systems to external information and data systems has been explored and developed. The technique is based on six identified data groups which together form the model with a context support. By using this technique, it is possible to update or replace only one type of data in the model, which provides a different characteristic to the model. It also extends the use of data in a new context since every piece of data is independent and can be used thereafter.

Specific industrial best practise contribution to knowledge for this research study includes:

- In current industrial practice, there is a prevailing one-to-one relationship between model and system. This thesis has demonstrated that it is possible to reuse data and models. It has also demonstrated that one model may be used as a basis to develop or customise a new one almost at runtime if the framework is used and the original model is designed according to the structure presented in this thesis.
- It has been demonstrated that simulation data and models can be reused in an off-the-shelf simulation tool that is not in itself equipped with input and output functionality for reconfiguration of model data. This can be achieved with support from existing standards such as STEP and XML. The use of information technology in general also facilitates the ability to connect and share processes and information irrespective of time and location.
- In conclusion; from an industrial perspective, this research has successfully advanced simulation data and model reusability in the area of virtual manufacturing with a focus on discrete event simulation. The research demonstrates the use of standards together with off-the-shelf simulation packages in a framework.

8.4 Future Work

The approach of this research investigation has been successfully tested within the experiments conducted. The research provides a framework that enables reuse of data across all the different applications that are used in virtual manufacturing.

Future research and development within this area could include the following:

- A natural next step based on the research presented in this thesis is to expand the tools used together with the framework presented. During the formulation of the framework some minor tests were carried out with a continuous path simulation program (IGRIP). These tests clearly indicated that the same integration technique and the format of the data suit this type of simulation tool. However, there were no full scale model experiments conducted with this type of tool since it was outside the scope of the investigation. Nevertheless, the tests indicate that it is possible to use the framework to support this type of simulation tool as well. Thus, many of the most common manufacturing simulation scenarios can be covered by the framework. Future research could involve more tests with computer aided robotic systems and also ergonomical studies. Similarly, tests with discrete event simulation tools from other vendors should also be carried out.

- The framework proposed in this thesis has been tested in one industrial manufacturing line. Although this test has been successful, how this framework performs in industrial environments in general, should be investigated. Particular topics of interest include the availability of data and information and its consequences. A typical question is whether, and how, too much data or inconsistent data can be handled by the framework.

References

[Andreasen, 1991]

Andreasen, M. M. (1991). The Theory of Domains. *In Proceedings of Workshop on Understanding Function and Function-to-Form Evolution.* Cambridge University

[Aganovic, 2004]

Aganovic, D., (2004) *On manufacturing system development in the context of concurrent engineering*, PhD thesis, Kungliga Tekniska Högskolan. ISSN 1650-1888

[Azari, 1990]

Azari, M (1990). *An integrated and Interactive Process Planning System Based on Product Model and AI.* Licentiate thesis, Kungliga Tekniska Högskolan. ISSN 0280-2074

[Bacha and Yannou, 2002a]

Bacha, R. and Yannou, B. (2002). A Methodology for Modeling Process Information - Towards a Process Technical Data Management. Chedmail P., Cognet G., Fortin C., Mascle C. and Pegna J. (Eds.) *In Integrated Design and Manufacturing in Mechanical Engineering.* Kluwer Academic Publishers: Dordrecht/Boston/London.

[Bacha and Yannou, 2002b]

Bacha, R. and Yannou, B. (2002). New Approach for Building an Integrated Information System for Manufacturing Engineering Departments. *MIM2000: IFAC Symposium on Manufacturing, Modeling, Management and Supervision*, Patras, Greece.

[Bacha and Yannou, 2002c]

Bacha, R. and Yannou, B. (2002). Guideline for Technical Data Integration in Manufacturing Engineering. *CE2000: 7th ISPE International Conference on Concurrent Engineering*, Lyon, France

[Balci, 1995]

Balci, O., (1995), Principles and techniques of simulation validation, verification and testing, *In proceedings of Winter Simulation Conference*, pp 147-154

[Ball, 1996]

Ball, P. D. (1999), Introduction to discrete event simulation, *In Proceedings of the 2nd DYCOMANS workshop on Management and Control: Tools in Action*, Algarve, pp 367-376]

[Banks et. al., 1996]

Banks, J., Carson, J. S. and Nelson, B. L., (1996), *Discrete-event system simulation*, 2nd edition, Prentice-Hall, Inc, Upper Saddler river, New Jersey 07458

[Banks, 2000]

Banks, J., (ed) (2000), *Handbook of simulation – Principles, Methodology, Advances, Applications, and Practice*, John Wiley & Sons, Inc.

[Bauer et. al., 1991]

Bauer, A., Bowden, R., Browne, J., Duggan, J., amd Lyons, G. (1991). *Shop Floor Control Systems – from Design to Implementation.* Chapman & Hall. ISBN 0-412-36040-3

[Bergman and Klefsjö, 1998]

Bergman, B. and Klefsjö, B. (1998). *Quality – from Customer Need to Customer Satisfaction.* Studentlitteratur. ISBN 91-44-46331-6

[Berio and Vernadat 1999]

Berio, G. and Vernadat, F. B., (1999), New developments in enterprise modelling using CIMOSA, *Computers in Industry*, Vol 40, pp 99-114.

[Bernard, 2000]

Bernard, B., (2000), *Integration of Manufacturing Simulation Tools with Information Sources*, Licentiate thesis, Kungliga Tekniska Högskolan. ISSN 1104-2125

[Bernus and Nemes, 1997]

Bernus, P. and Nemes, L. (1997) Requirements Of The Generic Enterprise Reference Architecture And Methodology, A. Rev. Control, 21, 125-136.

[Buckley, 1996]

Buckley, F. J., (1996), *Implementing configuration management: Hardware, software and firmware*. Second edition. IEEE Press

[Bolmsjö et. al., 1999]

Bolmsjö G, Lorentzon U, Randell L, Integration of development of products and production system using advanced simulation tools, *Proceedings of the 32nd CIRP International Seminar on Manufacturing Systems*, May 1999, Leuven,Belgium.

[Carrie, 1988]

Carrie, A., (1988), *Simulation of Manufacturing Systems*, John Wiley & Sons Ltd.

[CIMdata, 2001]

CIMdata, (2001), *Collaborative Product Definition Management : an overview*, Ann Arbor, MI

[Curtis, 1988]

Curtis, M. A. (1988). *Process Planning*. John Wiley & Sons. ISBN 0-471-83254-5

[Delmia, 2005]

Delmia (2005), accessed 25 April 2005, Available: www.delmia.com

[Doumeingts et. al., 1993]

Doumeingts, G., Chen, D., Vallespir, B. and Fenie, P., (1993), GIM (GRAI integrated methodology) and its evolutions a methodology to design and specify advanced manufacturing systems, *IFIP Transactions B: Computer Applications in Technology*, n B-14, pp 101-117

[Drake and Smith, 1996]

Drake, G. R. and Smith, J. S., (1996), Simulation system for real-time planning, scheduling and control, *In Charnes, Morrice, Brunner and Swain*, pp 1083-1090

[Eriksson, 1996]

Eriksson, P. (1996) *Enhancements in virtual robotics: Through simulation of sensors, events and 'pre-emptive' learning*, PhD Thesis, Faculty of Computing Sciences and Engineering, De Montfort University, Leicester.

[Eversheim et. al., 1991]

Eversheim, W., Marcziniski, G., and Cremer, R. (1991) Structured Modeling of Manufacturing Processes as NC-Data Preparation. *In Annals of the CIRP*, volume 40, pp 429-432

[Eversheim and Westekemper, 2001]

Eversheim, W. and Westekemper, M. (2001). Integrated Product and Process Model of SFB (Collaborative Research Center) 361. *Science & Research*, (2)

[Gogg and Mott, 1993]

Gogg, T. J. and Mott, J. R. A., (1993), Introduction to simulation, *In proceedings of the Winter Simulation Conference*, pp 9-17

[Gosling et. al., 2005]

Gosling, J. Joy, B., Steele, G. and Bracha, G., (2005) *The Java Language Specification.* Third edition, Addison-Wesley, ISBN 0-321-24678-0

[Grewal et. al., 1998]

Grewal, N. S., Bruska, A. C., Wulf, T. M. & Robinson, J. K. (1998), Integrating targeted cycle-time reduction into the capital planning process, *in Benjamin et al. (1998)*, pp. 1005–10 vol.2.

[Gunasekaran, 1999]

Gunasekaran, A. (1999) Agile manufacturing: A framework for research and development, *International Journal of Production Economics,* 62, pp. 87-105.

[Hicks et. al., 2002]

Hicks, B. J., Culley, S. J. Allen, R. D. and Mullineux, G., (2002), A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design. *International Journal of Information Management*, Vol. 22, pp 263-280

[Hitomi, 1979]

Hitomi, K. (1979). *Manufacturing System Engineering.* Taylor & Francis. ISBN 0-85066-177-3.

[Holmer et. al., 1990]

Holmer, B., Kjellberg, T., Fentorp, K.-O., Johansson, B. Stark, B. and Stenviken, B., (1990), *Verkstadsföretaget som informationssystem.* Styrelsen för Teknisk Utveckling. ISBN 91-7850-162-8

[Holst, 2001]

Holst, L. (2001). *Integrating Discrete-Event Simulation into the Manufacturing System Development Process.* Lic Thesis, Division of Robotics, Department of Mechanical Engineering, Lund University.

[Hubka and Eder, 1988]

Hubka, V. and Eder, W. E. (1988) *Theory of Technical Systems – A Total Concept Theory for Engineering Design.* Springer-Verlag. ISBN 3-540-17451-6

[Hull and King, 1987]

Hull, R. and King, R., (1987). Semantic database modelling: Survey, applications and research issues. *ACM Computing Surveys 19*, no. 3. pp 201–260.

[IFIP-IFAC, 1999]

IFIP-IFAC, (1999), Task Force, *GERAM Generalised Enterprise Reference Architecture and Methodology*, Version 1.6.3.

[ISO 8879, 1999]

Information processing – Text and office systems – Standard Generalized Markup Language (SGML), International Organisation for Standardisation, ISO 8879

[ISO/IEC, 1997]

The Virtual Reality Modelling Language, International Organisation for Standardisation, ISO 14772

[ISO/TC184/SC4, 1994a]

Industrial automation systems and integration — Product data representation and exchange—Part 11: Description methods: The EXPRESS language reference manual. International Organisation for Standardisation, ISO 10303-11

ISO/TC184/SC4, 1994b]

Industrial automation systems and integration — Product data representation and exchange—Part 203: Application protocol: Configuration controlled design, International Organisation for Standardisation, ISO 10303-203

[ISO/TC184/SC4, 1998]

Industrial automation systems and integration – Product data representation and exchange – Part 26: Implementation method: Interface definition language binding to the standard data access interface, International Organisation for Standardisation (DIS), ISO 10303-26

[ISO/TC184/SC4, 2001]

Industrial Automation System and Integration –Product Data Representation and Exchange – Part 214: Application Protocol: Core Data for Automotive Mechanical Design Processes. International Organisation for Standardisation, ISO 10303-214

[ISO/TC184/SC4, 2002]

Industrial automation systems and integration — Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure. International Organisation for Standardisation, ISO 10303-21

[ISO/TC184/SC4, 2005]

Industrial automation systems and integration manufacturing management data exchange MANDATE – Part 1: Overview and fundamental principles. ISO 15531

[Iwata et. al., 1995]

Iwata, K., Onosato, M., Teramoto, K. and Osaki, S., (1995), A Modelling and Simulation Architecture for Virtual Manufacturing Systems, *In Annals of the CIRP*, volume 44, pp 399-402

[Jacobsen et. al., 1992]

Jacobson, I., Christerson, M., Jonsson, P. and Vergaard, G., (1992), *Object-oriented Software Engineering: A Use Case Driven Approach*, ACM Press, New York, NY.

[Jain et. al., 2001]

Jain, S., Choong, N. F., Aye, K. M. and Luo, M. (2001), Virtual factory: an integrated approach to manufacturing systems modeling, *International Journal of Operations & Production Management*, Volume 21 pp. 594-608

[Johansson, 2001]

Johansson, M. (2001). *Information Management for Manufacturing System Development.* PhD thesis, Kungliga Tekniska Högskolan. ISSN 1650-1888

[Juran, 1988]

Juran, J. M. (1988). *Juran on Planning for Quality.* Free Press. ISBN 0-02-916681-0

[Juran and Gryna, 1988]

Juran, J. M. and Gryna, F. M., editors (1988). *Juran's Quality Control Handbook.* McGraw-Hill. ISBN 0-07-033176-6

[Jägstam, 2004]

Jägstam, M., (2004), *On efficient use and integration of discrete event simulation in industry*, PhD Thesis, Faculty of Computing Sciences and Engineering, De Montfort University, Leicester.

[Katz, 1990]

Katz, R. H., (1990), Toward a unified framework for version modeling in engineering databases. *ACM Computer Surveys* Vol. 22, no. 4 pp 375-408

[Kemmerer, 1999]

Kemmerer, S. J. (ed), (1999), *STEP The Grand Experience*, NIST Special Publication 939, Gaithersburg, MD 20899-0001.

[Kimura, 1993]

Kimura, F., (1993), Product and Process Modelling as a Kernal for Virtual Manufacturing Environment, *In Annals of the CIRP*, volume 42, pp 147-150.

[Kjellberg and Bohlin, 1996]

Kjellberg, T, and Bohlin, M., (1996), Design of a manufacturing resource information system, *In Annals of the CIRP - Manufacturing Technology*, volume 45, pp 149-152

[Kosanke, 1995]

Kosanke, K. (1995) CIMOSA - Overview and status, *Computers in Industry*, Vol 27, pp 101-109.

[Kosanke and Zelm 1999]

Kosanke, K. and Zelm, M., (1999), CIMOSA modelling processes, *Computers in Industry*, Vol 40, pp 141-153.

[Kosanke et. al., 1999]

Kosanke, K., Vernadat, F. B. and Zelm, M. (1999) CIMOSA: enterprise engineering and integration, *Computers in Industry*, Vol. 40, pp 83-97.

[Krause et. al., 1993]

Krause, F.-L., Kimura, F., Kjellberg, T., and Lu, S. C.-Y. (1993). Product Modelling. In *Annals of the CIRP*, pp 695-706. ISBN 3-905-277-20-4

[Kreutzer, 1986]

Kreutzer, W., (1986), *System simulation – Programming Styles and Languages*, Addison and Wesley, ISBN 0-201-12914-0

[Kulvatunyou and Wysk, 2000]

Kulvatunyou, B. and Wysk, R. A. (2000). A Functional Approach to Enterprise-Based Engineering Activities. *Journal of Manufacturing Systems*, 19(2)

[Law and Kelton, 1991]

Law, A. M. and Kelton, D. W., (1991), *Simulation modelling and Analysis*, Industrial Engineering and Management Science, 2nd edition, McGraw-Hill Inc.

[Law, 2003]

Law, A. M., (2003), How to conduct a successful simulation study, *In proceedings of Winter Simulation Conference*, pp 66-70

[Lee and Noh, 1997]

Lee, K. I. and Noh, S. D., (1997), Virtual Manufacturing System – A Test-bed of Engineering Activities, *In Annals of the CIRP*, volume 46, pp 347-350.

[Li and Williams, 2000]

Li, H. and Williams, T. J., (2000), Interconnected chain of enterprises as presented by the Purdue Enterprise Reference Architecture, *Computers in Industry*, Vol 42, pp 265-274.

[Lin et. al., 1995]

Lin, E., Minnis, I., Nau, D. and Regli, W. (1995), *Contribution to Virtual Manufacturing: Background Research*, report for Lawrence Associates Inc, Institute for Systems Research, University of Maryland.

[Liu and Xu, 2001]

Liu, D. T. and Xu, W., (2001), A review of web-based product data management systems, *Computers in Industry*, Vol. 44, pp 251-262

[Liyanage, 1999]

Liyanage, K. (1999). *Methodology For Input Data Modelling In The Simulation of Manufacturing Systems*, PhD Thesis. Sheffield Hallam University.

[Liyanage and Perera, 1998]

Liyanage, K. and Perera, T., (1998), Design and development of rapid data collection methodology, *In proceeding of international conference of simulation*, IEE pp 297-304

[Microsoft, 2005]

Microsoft (2005) *OLE*, Microsoft, accessed 12 September 2005, Available: www.microsoft.com

[Mills et. al., 2001]

Mills, J.J., Goossenaerts J.B.M. and Pels, H.J., (2001), The role of Context in the Product Realisation Process, *In proceedings of the CIRP Design Seminar*, pp. 175-180

[Nahavandi and Preece, 1994]

Nahavandi, S. and Preece, C., (1994), Virtual Manufacturing environment with an element of reality, *In proceedings of the 4th International Conference on Factory 2000 – Advanced Factory Automation*, pp 624-629

[Nance, 1994]

Nance, R. E., (1994), The conical methodology and the evolution of simulation model development, *Annals of Operations Research* vol. 54, pp 1–45

[Ng, 2003]

Ng, A. (2003) *An integrated design, simulation and programming environment for modular manufacturing machine systems*, PhD Thesis, Faculty of Engineering and Computing Sciences, De Montfort University, Leicester.

[Nielsen and Kjellberg, 2000]

Nielsen, J. and Kjellberg, T. (2000). The ISO 10303-214 Process Model as a Core for a Process Planning Tool. In *International CIRP Design Seminar*

[Nielsen, 2003]

Nielsen, J. (2003). *Information Modeling of Manufacturing Processes*. PhD thesis, Kungliga Tekniska Högskolan. ISSN 1650-1888

[NIIIP 1998]

NIIIP (1998) *Introduction to NIIIP Concepts*, NIIIP, accessed 27 April 2005, Available: www.niiip.org

[NIIIP 2001]

NIIIP (2001) *National Industrial Information Infrastructure Protocols - Project Summary*, NIIIP, accessed April 27 2005, Available: www.niiip.org

[Nordgren, 1995]

Nordgren, W. B., (1995), Steps for proper simulation project management, In *proceedings of Winter Simulation Conference*, pp 68-73

[Oh et. al., 2001]

Oh, Y., Han, S. and Suh, H., (2001), Mapping product structures between CAD and PDM systems using UML, *Computer-Aided Design*, Vol. 33, pp 521-529

[Olofsgård, 2005]

Olofsgård, P., (2005), *Distributed systems integration for development of modular machines*, PhD Thesis, Faculty of Engineering and Computing Sciences, De Montfort University, Leicester.

[OMG, 1997]

OMG, (1997), *The Common Object Request Broker: Architecture and Specification*, Revision 2.1, Object Management Group, ftp://ftp.omg.org/pub/docs/formal/97-09-01.pdf (Accessed 2005-04-25)

[Onosato and Iwata, 1993]

Onosato, M. and Iwata, K. (1993). Development of Virtual Manufacturing System by Integrating Product Models and Factory Models. *In Annals of the CIRP*, volume 42, pp 475-478.

[Ortiz et. al., 1999]

Ortiz, A., Lario, F. and Ros, L., (1999), Enterprise Integration—Business Processes Integrated Management: a proposal for a methodology to develop Enterprise Integration Programs, *Computers in Industry*, Vol 40, pp 155-171.

[Oscarsson, 2000]

Oscarsson, J. (2000) *Enhanced Virtual Manufacturing: advanced digital mock-up technology with simulation of variances*, PhD Thesis, Faculty of Engineering and Computing Sciences, De Montfort University, Leicester.

[Oscarsson and Urenda Moris 2002]

Oscarsson J. and Urenda Moris M. (2002) Documentation of discrete event simulation models for manufacturing system life cycle simulation, *In Proceedings of Winter Simulation Conference*, pp 1073-1078

[Oxford English Dictionary, 2002]

Simpson, J., A, (Edt), (2002), *Oxford English Dictionary*, Oxford University Press, ISBN - 0-1952-1889-2)

[Parks et. al., 1994]

Parks, C. M., Koonce, D. A., Rabelo, L. C., Judd, R. P. and Sauter, J. A., (1994), Model-based manufacturing integration: A paradigm for virtual manufacturing systems engineering, *Computers and Industrial Engineering* 27 (1-4), pp 357 – 360

[Paul and Taylor, 2002]

Paul, R. and Taylor, S., (2002), What use is model reuse: is there a crook at the end of the rainbow?, *In Proceedings of Winter Simulation Conference*, pp 648-652

[Peltonen, 2000]

Peltonen, H., (2000). *Concepts and Implementation for Product Data Management.* PhD thesis. Helsinki University of Technology, ISSN 1456-9418

[Peters et. al., 1996]

Peters, B. A., Smith, J. S., Curry, J., LaJimodiere, C. and Drake, G. R., (1996), Advanced tutorial-simulation-based scheduling and control, In Charnes, Morrice, Brunner and Swain, pp. 194-198

[Pidd, 2002]

Pidd, M., (2002), Simulation software and model reuse: a polemic, *In Proceedings of the 2002 Winter Simulation Conference*, pp 772-774

[QUEST, 2005]

QUEST (2005) QUEST, Ver 5.12SP9, DELMIA, Dassault Systemes

[Reese and Wyatt, 1987]

Reese, R. and Wyatt, D.L., (1987), Software reuse and simulation, in: A. Thesen, H. Grant, W. D. Kelton (Eds.), *Proceedings of the 1987 Winter Simulation Conference*, 14–16 December 1987 SCS, San Diego, CA.

[Robertson and Perera, 2002]

Robertson, N. and Perera, T.(2002). Automated data collection for simulation?. *Simulation Practise and Theory*, Vol. 9, pp 349-364.

[Robinson et. al., 2004]

Robinson, S., Nance, R., Paul, R., Pidd, M. and Taylor, S., (2004), Simulation model reuse: definitions, benefits and obstacles, *Simulation Modelling Practise and Theory*, Vol. 12, pp 479-494

[Ross and Schoman, 1977]

Ross, D. T. and Schoman, Jr., K. E., (1977). Structured Analysis for Requirements Definition. IEEE *Transactions on Software Engineering*, SE-3(1):6-15

[Sadowski, 1993]

Sadowski, R., (1993), Selling Simulation and Simulation Results, *In proceedings of Winter Simulation Conference*, Los Angeles pp 65-68

[Sadowski and Grabau, 2000]

Sadowski, D. A. and Grabau, M. R., (2000), Tips for successful practice of simulation, *In proceedings of Winter Simulation Conference*, pp 26-31

[Sargent, 1998]

Sargent, R. G. (1998), Verification and Validation of Simulation Models, *In proceedings of Winter Simulation Conference*, Washington D. C. pp 121-130

[Schenck and Wilson, 1994]

Schenck, D. A. and Wilson, P. R., (1994), *Information modelling: The EXPRESS way.* Oxford University Press. ISBN 0-8186-2088-9

[Segarra, 1999]

Segarra, G., (1999), The advanced information technology innovation roadmap, *Computers in Industry*, Vol 40, pp 185-195.

[Shukla et. al., 1996]

Shukla, C., Vazquez, M. and Chen, F., (1996) Virtual Manufacturing: An Overview, *Journal of Computers in Industrial Engineering*, Vol 31, No 1 pp. 79-82

[Sommerville, 2001]

I. Sommerville, I., (2001), *Software Engineering*, 6th Edition., Pearson Education Limited, Harlow.

[Steiner et. al., 2002]

Steiner, S. J., de Vicq, A. and Medland, A. J., (2002), ARMMS – Agility and Re-Configurability – Drivers, Approaches and Classification Methods, *WMC 2001*, Rochester, NY.

[Szykman et. al., 1999]

Szykman S., Senfaute, J. and Sritam, R.D., (1999), Using xml to describe function and taxonomies in computer-based design. *In Proceedings of the ASME Engineering Technical Conference*, pp 1-11

[Szyperski, 1999]

Szyperski, C., (1999), *Component Software: Beyond Object-oriented Programming*, Pearson Education Limited, Harlow.

[Swain, 1999]

Swain, J. J., (1999), *Imagine new worlds*, OR/MS Today, Available online from http://www.lionhrtpub.com/orms/orms-2-99/sim.html [accessed April 20 2005]

[Teeuw et. al., 1996]

Teeuw, W. B., Liefting, J. R., Demkes, R. H. J. and Houtsma, M. A. W., (1996), Experiences with product data interchange: On product models, integration, and standardisation. *Computers in Industry,* Vol. 31, no. 3, pp 205–221.

[Tian et. al., 2002]

Tian, G.Y., Yin, G. and Taylor, D., (2002), Internet-based Manufacturing: A review and a new infrastructure for distributed intelligent manufacturing, *Journal of Intelligent Manufacturing*, 13:323-338.

[Trybula, 1994]

Trybula, W., (1994), Building simulation models without data, In IEEE International Conference on Systems, Man and Cybernetics. Humans, *Information and Technology*, Vol. 1 IEEE pp 209-214

[Tönschoff and Zwick, 1998]

Tönschoff, H. K. and Zwick, M. (1998). An Integrated Product and Process Model. *In Globalization of Manufacturing in the Digital Communications Era of the 21st Century*, pp 209-219. Kluwer Academic Publisher.

[Ulrich and Eppinger, 2000]

Ulrich, K. T. and Eppinger, S. D. (2000). *Product Design and Development.* McGraw-Hill, second edition. ISBN 0-07-116993-8

[Upton and McAfee, 1996]

Upton, D. M. and McAfee, A. P., (1996), The real virtual factory, *Harvard Business Review*, July-August

[Urenda et. al., 2004]

Urenda Moris M., De Vin L.J. and Eriksson P. (2004) Introducing discrete event simulation for decision support in the Swedish health care system, *In Proceedings of the 2004 Western MultiConference: Health Sciences Simulation 2004* (Eds. Anderson J.G. and Katzper M.), San Diego CA:SCS-Society for Modeling and Simulation International, San Diego, CA., pp. 48-53.

[van den Hamer and Lepoeter, 1996]

van den Hamer, P. and Lepoeter, K., (1996), Managing designs data: The five dimensions of CAD frameworks, configuration management, and product data management. *In proceedings of the IEEE.* vol. 84, no. 1, pp 42-56

[van der Aalst and van Hee, 2002]

van der Aalst, W. and van Hee, K., (2002) *Workflow management – models, methods and systems*, MIT press, Cambridge, Massachusetts, London, ISBN 0-262-01189-1

[Vernadat, 1996]

Vernadat F.B. (1996) *Enterprise Modelling and Integration*, Chapman & Hull, ISBN 0-412-60550-3

[Weston, 1993]

Weston, R.H. (1993) Steps towards enterprise-wide integration: A definition of needs and first generation open solutions. *International Journal of Production Research*, Vol 31, pp 2235-2254.

[Williams, 1994]

Williams, T. J., (1994), Purdue Enterprise Reference Architecture, *Computers in Industry*, Vol 24, pp 141-158.

[Young et. al. 2000]

Young, B., Canciglieri, Jr, O., Costa, C. A., Dorador, J. M., Zhao, J., and Cheung, W. M. (2000). Information Support in an Integrated Product Development System. In *Integrated Design and Manufacturing in Mechanical Engineering.*